

**UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN**  
**FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA**



**EVALUACIÓN DEL PLANIFICADOR AUTOMATIZADO OPTIC AL  
GENERAR RUTAS DE APRENDIZAJE MEDIANTE EL USO DE  
COTAS SUPERIORES Y EXTENSIÓN DEL MODELO MATEMÁTICO**

**POR**

**JUAN PABLO ROSAS BALDAZO**

**EN OPCIÓN AL GRADO DE MAESTRÍA  
EN CIENCIAS EN INGENIERÍA DE SISTEMAS**

**OCTUBRE, 2018**

**UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN**  
**FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA**  
**SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO**



**EVALUACIÓN DEL PLANIFICADOR AUTOMATIZADO OPTIC AL  
GENERAR RUTAS DE APRENDIZAJE MEDIANTE EL USO DE  
COTAS SUPERIORES Y EXTENSIÓN DEL MODELO MATEMÁTICO**

**POR**

**JUAN PABLO ROSAS BALDAZO**

**EN OPCIÓN AL GRADO DE MAESTRÍA  
EN CIENCIAS EN INGENIERÍA DE SISTEMAS**

**SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN, MÉXICO**

**OCTUBRE 2018**

**Universidad Autónoma de Nuevo León**  
**Facultad de Ingeniería Mecánica y Eléctrica**  
**División de Estudios de Posgrado**

Los miembros del Comité de Tesis recomendamos que la Tesis «Evaluación del planificador automatizado OPTIC al generar rutas de aprendizaje mediante el uso de cotas superiores y extensión del modelo matemático.», realizada por el alumno Juan Pablo Rosas Baldazo, con número de matrícula 1266973, sea aceptada para su defensa como opción al grado de Maestría en Ciencias en Ingeniería de Sistemas.

El Comité de Tesis

---

Dr. Romeo Sánchez Nigenda

Asesor

---

Dra. Cristina Maya Pádrón

Revisor

---

Dra. Edith Lucero Ozuna Espinosa

Revisor

Vo. Bo.

---

Dr. Simón Martínez

División de Estudios de Posgrado

San Nicolás de los Garza, Nuevo León, octubre 2018

# ÍNDICE GENERAL

---

<b>Agradecimientos</b>	<b>XI</b>
<b>Resumen</b>	<b>XII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Descripción del Problema de Planificación Educativa . . . . .	2
1.2. Objetivos . . . . .	5
1.3. Estructura de la tesis . . . . .	5
<b>2. Marco Teórico</b>	<b>7</b>
2.1. Relevancia del Problema . . . . .	7
2.2. Planificación Automatizada . . . . .	8
2.2.1. PDDL . . . . .	12
2.3. OPTIC . . . . .	15
2.3.1. Introducción a OPTIC . . . . .	15
2.3.2. Representación de planes mediante grafos . . . . .	17
2.3.3. Búsquedas más comunes . . . . .	21
2.3.4. Resolviendo un Problema con OPTIC . . . . .	24

---

<b>3. Metodología</b>	<b>28</b>
3.1. Descripción del constructivo . . . . .	29
3.2. Obtención de Resultados con OPTIC . . . . .	32
3.3. Obtención de el valor óptimo . . . . .	33
3.4. Extensión del Modelo matemático del PPE . . . . .	34
3.4.1. Restricciones para obtener una secuencia . . . . .	35
3.4.2. Modelación del Estrés . . . . .	36
3.4.3. Modelación de la calidad del material . . . . .	37
3.4.4. Complejidad de las actividades . . . . .	38
3.4.5. Modelación de la capacidad del estudiante . . . . .	38
3.4.6. Modelo Matemático Completo . . . . .	39
<b>4. Experimentación</b>	<b>41</b>
4.1. Descripción de las instancias . . . . .	42
4.2. Experimentación 1 . . . . .	43
4.2.1. Resultados 1 . . . . .	44
4.3. Experimentación 2 . . . . .	51
4.3.1. Resultados 2 . . . . .	52
<b>5. Conclusiones</b>	<b>55</b>
5.1. Contribuciones . . . . .	55
<b>6. Trabajo Futuro</b>	<b>57</b>

---

<b>A. Ejemplos</b>	<b>59</b>
A.1. PDDL . . . . .	59
A.1.1. Problema PDDL . . . . .	60
A.1.2. Dominio PDDL . . . . .	63

# ÍNDICE DE FIGURAS

---

1.1. Ejemplo de Estructura de un Problema de Planificación Educativa. . .	4
2.1. Ejemplo de un Grafo de Estados. . . . .	11
2.2. Función Objetivo del <i>PPE</i> en PDDL. . . . .	13
2.3. Condiciones iniciales de variables para el PPE en PDDL. . . . .	13
2.4. Parte del dominio de un PPE en PDDL. . . . .	14
2.5. Instanciación de objetos de un PPE en PDDL. . . . .	15
2.6. Ejemplo de una Acción Temporal dividida en dos, una de inicio y una de fin. La etiqueta <i>al</i> hace referencia al <i>makespan</i> acumulado. . . . .	18
2.7. Parte de un grafo TRPG de un Problema de Planificación Educativa. . .	19
2.8. Representación gráfica de un Grafo de Estados de <i>OPTIC</i> . . . . .	21
2.9. Imagen para Algoritmo Ascenso de Colinas . . . . .	23
3.1. Diagrama de Flujo del CSA. . . . .	31
4.1. Comparación de resultados obtenidos con el <i>CSA</i> y <i>OPTIC</i> contra el valor Óptimo, separados por la cantidad de materias. . . . .	47
4.2. Comparativa de tiempos de computo entre <i>OPTIC</i> y el constructivo. . .	48

---

4.3. Tiempo de computo del constructivo al aumentar la cantidad de soluciones a generar. . . . .	49
4.4. Gap promedio del constructivo y el planificador OPTIC. . . . .	50
4.5. Gap promedio en caja de bigotes. . . . .	51
4.6. Instancias en las que se mejoro la solución encontrada con <i>OPTIC</i> utilizando soluciones iniciales proporcionadas por el <i>CSA</i> (puntos en color rojo). . . . .	52
4.7. Gap promedio. . . . .	54



# ÍNDICE DE TABLAS

---

2.1. Primer solución encontrada por el planificador OPTIC. . . . .	26
2.2. Primer solución encontrada por el planificador OPTIC. . . . .	26
3.1. Formato de los datos extraídos de un problema de ejemplo. . . . .	32
4.1. Cantidad de problemas por tipo de instancia según la cantidad de materias. La cantidad de temas es por materia, igual sucede con los subtemas, la cantidad es por cada tema y lo mismo las actividades, es la cantidad de actividades por subtema, al final se muestra el total de actividades por tipo de instancia. . . . .	42
4.2. Tipos de instancias según su dependencia, los valores en las celdas corresponden a porcentaje de actividades, El tipo <i>A</i> hace referencia a que solo requiere una actividad como requisito, tipo <i>B</i> , requiere dos actividades, la columna de obligatorias hace referencia al porcentaje de actividades que se tienen que incluir en la secuencia. . . . .	43
4.3. Cantidad de instancias resueltas por <i>OPTIC</i> . . . . .	44
4.4. Cantidad de instancias que mejoraron sus soluciones <i>OPTIC</i> . . . . .	53

# AGRADECIMIENTOS

---

Aquí puedes poner tus agradecimientos. (No olvides agradecer a CONACyT en caso de que hallas sido beneficiado con una beca).

# RESUMEN

---

Juan Pablo Rosas Baldazo.

Candidato para el grado de Maestro en Ingeniería  
con especialidad en Ingeniería de Sistemas.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Título del estudio:

## EVALUACIÓN DEL PLANIFICADOR AUTOMATIZADO OPTIC AL GENERAR RUTAS DE APRENDIZAJE MEDIANTE EL USO DE COTAS SUPERIORES Y EXTENSIÓN DEL MODELO MATEMÁTICO.

Número de páginas: 70.

**OBJETIVOS Y MÉTODO DE ESTUDIO:** En este trabajo se presentará un algoritmo constructivo para el problema de planificación educativa (PPE) con el fin de demostrar que al proporcionar soluciones iniciales de buena calidad a un planificador automatizado, en este caso al planificador *OPTIC*, se puede alcanzar esa misma solución o una mejor en menos tiempo.

También se agregaron nuevas restricciones al modelo matemático del problema de planificación educativa (PPE) las cuales consideran características subjetivas

como el estrés, la capacidad intelectual y la calidad del material, esto con el fin de que el problema sea más apegado a la realidad. También se generaron restricciones que le permiten al modelo generar una secuencia en las actividades.

CONTRIBUCIONES Y CONCLUSIONES: Desarrollo de un constructivo greedy el cual genera soluciones de forma aleatoria para el PPE, el funcionamiento es similar al algoritmo Monte Carlo.

Se evaluó el desempeño del planificador *OPTIC* utilizando soluciones iniciales como cota para la búsqueda de una solución al *PPE*.

Se agregaron nuevas restricciones al modelo matemático para que considere características como estrés, capacidad intelectual y calidad del material, apegando el problema al mundo real. Al incluir este tipo de características en el *PPE* nos permite personalizar la ruta de aprendizaje de un usuario en particular, que es una de las principales áreas de investigación en el campo de educación asistida. Una vez construido el modelo matemático también nos permitiría incluir las nuevas restricciones en *PDDL* para ser resuelto con los planificadores.

Firma del asesor: \_\_\_\_\_

Dr. Romeo Sánchez Nigenda

## CAPÍTULO 1

# INTRODUCCIÓN

---

En la vida diaria la *planificación* se puede definir como el proceso de buscar un grupo de acciones que, ejecutadas en una secuencia en particular, permita satisfacer ciertas metas.

Dentro del ámbito educativo, la planificación de actividades que guíen al estudiante durante su vida educativa, tiene muchas áreas de oportunidad y es un tema de investigación, ya que la generación de este tipo de rutas debe ser personalizada para cada tipo de estudiante.

Es por ello que el uso de agentes inteligentes como los *planificadores automatizados* (descritos en la sección 2.2) en este tipo de problemas, nos permitiría obtener una ruta de actividades a seguir y satisfacer ciertos objetivos del problema.

Para la generación de soluciones, los planificadores automatizados pueden considerar ciertos aspectos definidos en el problema para la búsqueda de una solución, sin embargo, el número de estados en el espacio de búsqueda suele ser muy grande provocando que el tiempo y recursos de computo se incrementen, por lo que recurren a métodos heurísticos para la búsqueda de una solución, esto hace que no sea posible asegurar el valor óptimo.

El problema de generación de rutas de aprendizaje se ha resuelto anteriormente usando planificadores [9], sin embargo, como se menciono, a pesar los resultados obtenidos, la manera en que los planificadores buscan una solución puede mejorarse dado que las soluciones encontradas no son óptimas.

En este trabajo nos enfocaremos en realizar un análisis para encontrar los límites para los cuales un planificador temporal como *OPTIC* es capaz de encontrar una solución para el problema de planificación educativa, tratando de acotar el área hasta la cual la heurística de búsqueda le permite llegar al planificador, y comparándola con el valor óptimo del problema.

Se presenta también un algoritmo constructivo el cual se encarga de generar soluciones factibles las cuales servirán como solución inicial para el planificador, esto con el fin de acotar el espacio de búsqueda. El constructivo tiene características similares al algoritmo Monte Carlo, es decir genera un gran número de soluciones de las cuales se selecciona la mejor, esperando que a mayor número de soluciones generadas, se pueda obtener una de mejor calidad.

También se presentan nuevas restricciones en el modelo matemático del problema (sección 3.4) que nos permitan manejar características como estrés, capacidad intelectual y calidad del material con el que se trabaja en una determinada materia, lo cual impacta en la selección de actividades generando soluciones más personalizadas.

## 1.1 DESCRIPCIÓN DEL PROBLEMA DE PLANIFICACIÓN EDUCATIVA

El problema ya ha sido descrito anteriormente por *Maya P. et al.*, en el artículo "*Generación de trayectorias de aprendizaje a través de modelos educativos de planificación*" [9] donde se describe un modelo matemático del problema y se muestran algunos resultados usando algunos planificadores como *POPF* y *LPG*, por ello, de ahora en adelante nos referiremos al problema por las siglas *PPE* (*Problema de Planificación Educativa*), que fue la manera en como se nombro.

El problema se podría describir de la siguiente manera; dado un conjunto de materias las cuales tienen asociados temas, los temas tiene asociados subtemas y

los subtemas tienen actividades, y en donde cada actividad tiene un valor y una duración asignado, por lo que al ser ejecutadas, la suma de los valores de cada actividad seleccionada debe satisfacer una calificación mínima y una máxima.

El punto de interés del problema, es obtener que conjunto de actividades debe realizar un estudiante en una secuencia en particular con el fin de satisfacer ciertos objetivos. Además se busca minimizar el tiempo invertido para satisfacer dicha calificación.

En otras palabras, se busca minimizar el tiempo necesario para obtener una calificación aprobatoria al realizar, en una secuencia en particular, un grupo de actividades con una duración y valor asociado, la calificación o métrica mínima y máxima a alcanzar es definida según las preferencias del usuario.

Una representación gráfica del problema sería como se observa en la figura 1.1, cada actividad tiene asociado una duración y un valor, y donde la selección de actividades repercute directamente en la calificación del subtema, la calificación del tema y al final en la materia.

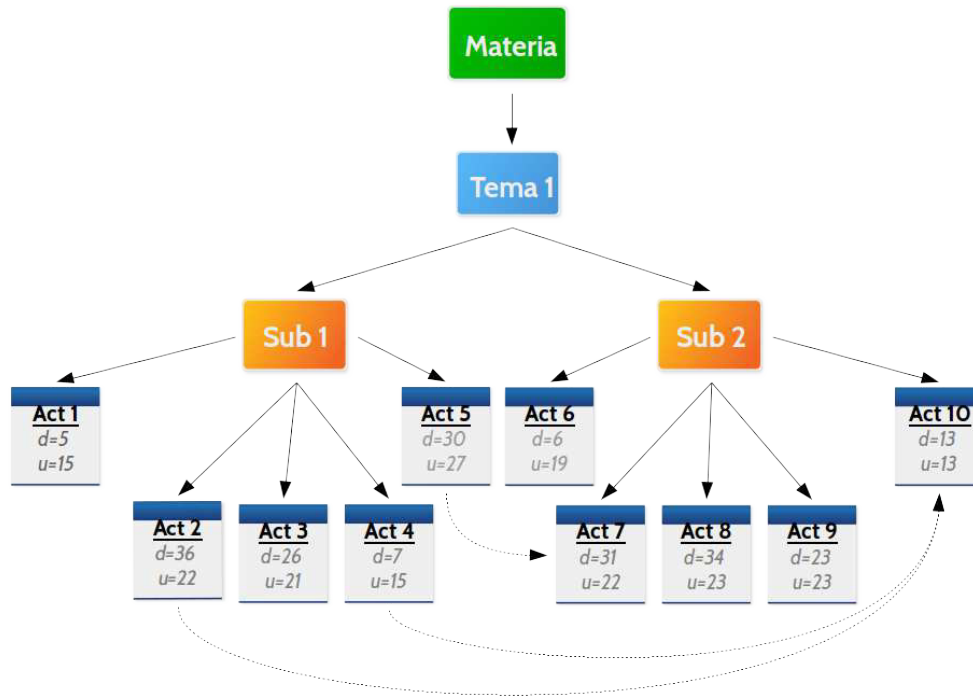


Figura 1.1: Ejemplo de Estructura de un Problema de Planificación Educativa.

El *PPE* presenta algunas restricciones las cuales se describen a continuación:

- Una de ellas son **restricciones de precedencia** o **restricciones de habilitamiento**, esto quiere decir que existen ciertas actividades que requiere que una o más actividades se realicen primero para poder ser seleccionadas.
- También existen **actividades obligatorias** las cuales deben de ser incluidas en la solución siempre, en algunos casos estas actividades tienen inclusive *actividades de habilitamiento* por lo que aparte de agregar las actividades obligatorias a la solución, se deben agregar las actividades requisito de estas.
- Otra restricción del problema es que el puntaje obtenido en cada subtema, tema o materia no debe sobrepasar un valor mínimo y máximo establecido por el usuario.



## 1.2 OBJETIVOS

Considerando resolver el *PPE* mediante planificadores automatizados, al incrementar la cantidad de actividades, el *grafo de estados* crece exponencialmente lo que causa que el planificador demore más tiempo para encontrar una solución.

Por tal motivo el primer objetivo de este trabajo es la generación de soluciones de buena calidad para el *PPE* mediante un constructivo greedy las cuales servirán para demostrar la hipótesis 1.

**Hipótesis 1.** - Al proporcionarle al planificador una solución inicial, se podría obtener una solución de igual o mejor calidad en menos tiempo, a diferencia de la manera en como el planificador busca su propia solución inicial.

El constructivo es similar al algoritmo *Monte Carlo* ya que se espera que la calidad de las soluciones mejore dependiendo de la cantidad de soluciones generadas.

Un segundo objetivo de la tesis es extender el modelo matemático del problema ya que actualmente no considera características subjetivas como estrés o la capacidad del estudiante, las cuales afectan en el desempeño del mismo, por lo que considerar este tipo de restricciones vuelve el problema más apegado a la realidad.

## 1.3 ESTRUCTURA DE LA TESIS

Durante los siguientes capítulos se describirá de manera breve lo concerniente al *PPE* que será el problema con el que se trabajará explicando el modelo matemático inicial y cual fue la parte que se extendió.

Se dará una breve introducción a la planificación automatizada en la sección 2.2, en particular lo que es más relevante para el planificador *OPTIC* el cual se describe en la sección 2.3.1. Se abordaran los tipos de grafos utilizados por el planificador *OPTIC* como el *TRPG* descrito en la sección 2.8 y se describirá la forma

en que la heurística de *OPTIC* selecciona y expande los nodos al momento de estar buscando una solución.

También se describirá en el capítulo 3 como funciona el algoritmo constructivo y en el *capítulo 4* se podrán observar los resultados obtenidos así como la descripción de las instancias utilizadas tanto en *OPTIC* como en el constructivo y una comparación de como el uso de una cota superior podría mejora el tiempo en que se encuentra una solución.

## CAPÍTULO 2

# MARCO TEÓRICO

---

Con el fin de obtener soluciones para el *PPE* haremos uso de una rama de la *Inteligencia Artificial* conocida como *planificación automatizada*, la cual sera descrita en las siguientes secciones al igual que detalles sobre el problema a trabajar.

## 2.1 RELEVANCIA DEL PROBLEMA

En algunos estudios se ha comprobado que al generar rutas de aprendizaje al estudiante durante su vida académica se obtiene mejores resultados en lugar de permitirle seleccionar que actividades realizar [7] .

La generación de rutas de aprendizaje la podemos definir como el proceso en el que dada una retícula educativa, se busca encontrar una secuencia de actividades que maximice el conocimiento adquirido por un aprendiz, guiándolo durante su vida educativa. Estas rutas generalmente tienen que ser personalizadas ya que no todos los individuos tienen el mismo conocimiento base o las mismas capacidades [6].

Este tipo de problemas es una de las más importantes investigaciones para algunos sistemas inteligentes como lo son los sistemas de aprendizaje basados en *web* (*WBE Web Based Education*) [17], que son sistemas orientados a estudiantes a distancia, pero que a pesar de ser muy buenos, presentan algunas desventajas dependiendo del tipo de estudiante que los utilice, el material y la disposición para aprender.

Otro ejemplo de sistemas inteligentes son los *ISA (Intelligent Social Agents)* [18] dado que generalmente las rutas encontradas no son apropiadas para todos los aprendices ya que se necesita tener en cuenta factores humanos como habilidades y capacidades del aprendiz.

Para este trabajo haremos uso de planificadores automatizados, que a diferencia de los dos sistemas mencionados anteriormente, no son diseñados específicamente para tratar con problemas educativos. Sin embargo, se adaptan perfectamente al problema dado que la finalidad del problema es encontrar una secuencia que satisfice una serie de objetivos.

Algunos aspectos del problema a considerar son una combinación de factores personales, factores de comportamiento y factores de conocimiento los cuales afectan la manera en que un estudiante aprende. Aunado a eso, existen también diferentes formas o estilos en las que un estudiante aprende. De acuerdo a esto los estudiantes tienen material el cual se debe ajustar a los estilos de aprendizaje para que aprendan más efectivamente y tengan un mejor progreso [11].

Los mejores resultados encontrados en esta clase de sistemas inteligentes son en personas con un mayor grado académico o estudiantes muy motivados, mientras que en personas con estudios iniciales se requiere más atención por parte de un profesor.

Algo que tienen en común estos sistemas inteligentes es la necesidad de caracterizar de alguna manera aspectos humanos para poder adaptar la manera en que se generara su guía de aprendizaje.

## 2.2 PLANIFICACIÓN AUTOMATIZADA

La *planificación* se puede definir como el proceso de búsqueda de un conjunto de acciones, que colocadas en una secuencia en particular, permiten satisfacer o cumplir ciertas metas a partir de un estado inicial.

Dentro del contexto de *Inteligencia Artificial* como describen *Russell S. y P.*

*Norving* [8, 16, 19], la planificación estudia el diseño de una secuencia de acciones que permiten a partir de un estado inicial, el alcance de un conjunto de objetivos.

Los algoritmos de planificación permiten no solamente encontrar el estado solución que satisface los requerimientos del usuario, sino también los estados intermedios que resultan de la ejecución de las acciones que forman el plan.

Los problemas de planificación tienen aplicaciones directas en situaciones del mundo real tales como turismo, misiones espaciales, industria, procesos de enseñanza/aprendizaje, robótica, transporte, etc., en este trabajo nos enfocaremos en procesos de enseñanza/aprendizaje.

La idea básica de los planificadores es dotar a un sistema inteligente de una representación de los objetivos a alcanzar, un conjunto de acciones posibles de aplicar, un entorno en el cual se va a desarrollar y la capacidad de poder generar un plan.

Para las siguientes secciones sera necesario introducir algunos conceptos utilizados comúnmente para describir procesos de planificación.

**Definición 2.2.1. Estado.** Un estado  $s$  es un conjunto finito de átomos lógicos. Generalmente en planificación temporal un estado se define como una tupla  $(F, V, Q, P, C)$ , donde  $F$  es un conjunto de proposiciones soportadas,  $V$  es un vector de variables numéricas, en algunos planificadores donde algunas acciones puede tener efectos numéricos continuos  $V$  es representada por una  $V^{min}$  y  $V^{max}$  la cota inferior y superior por cada  $V$ ,  $Q$  una lista de acciones que se iniciaron pero no han terminado, donde cada  $(a, s) \in Q$ , donde  $a$  identifica la acción y  $s$  el paso en el cual comienza,  $P$  es el plan de alcanzar el estado actual,  $C$  es una lista de restricciones temporales sobre el paso en  $P$ .

**Definición 2.2.2. Accion.** Una acción  $a$  es una tripleta de la forma  $a = (head(a), pre(a), eff(a))$  o de la forma  $a = (head(a), pre(a), eff(a), cost(a))$ , donde

- $head(a)$  es una función de la forma  $act(z_1, z_2, \dots, z_k)$  donde  $act$  representa el

nombre de la acción y  $z_1, z_2, \dots, z_k$  todas las variables numéricas necesarias en  $pre(a)$ .

- $pre(a) = \{p_1, p_2, \dots, p_m\}$  es un conjunto de precondiciones.
- $eff(a) = \{e_1, e_2, \dots, e_n\}$  es un conjunto de efectos de la forma  $e_n = sv(t_1, t_2, \dots, k_j) \leftarrow t_0$  donde  $t_k$  es el objetivo del efecto y  $t_0$  es el valor que se le va a asignar.
- $cost(a)$  es un número  $c > 0$  que representa el costo de aplicar la acción  $a$ , si el costo es omitido, el valor por defecto es  $cost(a) = 1$ .

**Definición 2.2.3. Problema de Planificación.** Un problema de planificación  $P$  se define como  $P = (A, I, G)$  donde  $A$  es un conjunto de acciones,  $I$  el estado inicial y  $G$  un conjunto de objetivos.

**Definición 2.2.4. Plan.** Dada un problema  $P = (A, I, G)$ , un plan es una secuencia  $p = (a_1, a_2, a_3, \dots, a_n$  de acciones de  $A$  que resuelven el problema de planificación  $P$ .

Tomando en cuenta estos conceptos, los problemas de planificación se pueden ver como una búsqueda dentro de un grafo de estados guiados por procedimientos heurísticos, como se observa en la figura 2.1, al llegar al estado *S38* se pueden satisfacer algunos objetivos, como lo es que el *Sub1* tenga una calificación aprobatoria. Al seguir avanzando, al llegar al estado *S97*, se logran satisfacer todos los objetivos por lo que es un estado solución, que puede o no ser una solución óptima.

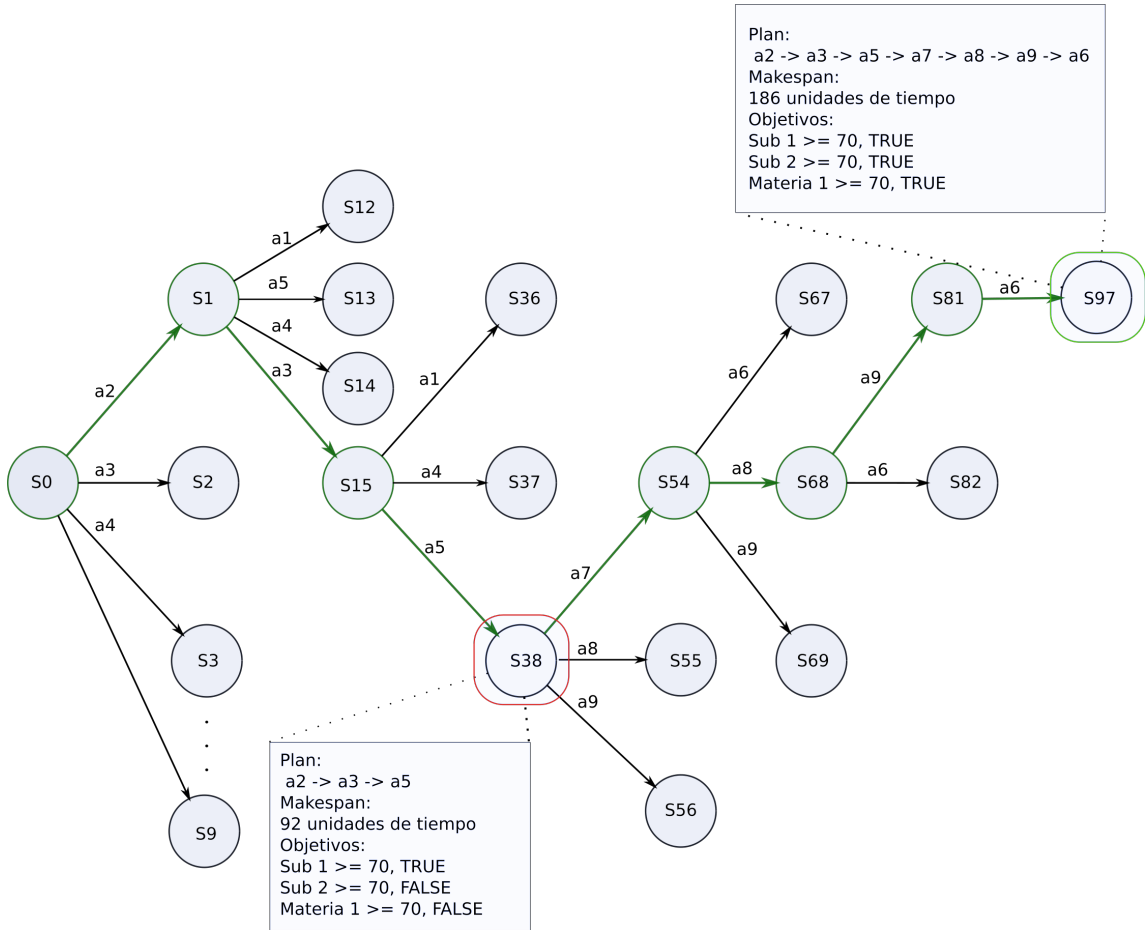


Figura 2.1: Ejemplo de un Grafo de Estados.

Es por eso que en base a lo antes descrito es de gran importancia generar heurísticas que además de otorgarnos soluciones de alta calidad nos proporcionen información útil para guiar la búsqueda y sean capaz de hacerlo en un corto tiempo.

Para poder realizar todo lo descrito, los planificadores requieren una manera de representar su entorno, las posibles acciones a aplicar, los estados objetivo y la capacidad de generar la secuencia que conformara el plan, todo esto lo logran mediante un lenguaje en común, en nuestro caso se conoce como *PDDL* (*Planning Domain Definition Language*) descrito en la sección 2.2.1.

Para este trabajo se utilizó un planificador temporal llamado *OPTIC* (*Optimizing Preferences and Time-dependent Costs*) que implementa características de

*planificación parcial* para darle algo de flexibilidad a la búsqueda.

Otro punto a considerar en *OPTIC* es que maneja las penalizaciones de forma continua por lo que incurrir en alguna penalidad, se considera solo una porción de la penalización.

### 2.2.1 PDDL

Motivados por la necesidad de un lenguaje de modelación capaz de expresar propiedades temporales y numéricas de diferentes tipos de dominios de problemas, en 1998 durante la *competencia internacional de planificación (ICAPS)* se presento un estándar de codificación llamado *PDDL (Planning Domain Definition Language)* el cual fue desarrollado por *Craig A. K.* [15] y por el comité internacional de Planificación, facilitando el desarrollo científico de la planificación.

Gracias a este estándar es que se ha logrado avanzar en el área de planificación automatizada, ya que permite expresar de forma clara modelos de dominios y así poder ser compartidos por la comunidad.

*PDDL* es un lenguaje centrado en acciones, es una estandarización de la sintaxis para expresar la semántica de acciones usando pre condiciones y post condiciones, la aplicación y los efectos de las acciones. La sintaxis se inspira en *LISP* así que mucha de la estructura es muy similar.

Un primer cambio en el diseño de *PDDL* es la representación de una aplicación en dos estructuras: el dominio y el problema. El dominio describe las acciones, sus condiciones de aplicabilidad y los efectos de las mismas en el ambiente. El problema describe un escenario (instancia) en términos de las condiciones iniciales del ambiente y los objetivos.



```

;;objetivo: termine todas las materias
(:goal (and
      (pass-degree Material1 student1)
    )
)

;; metrica a optimizar
(:metric minimize (total-time))

;;fin del problema
)

```

Figura 2.2: Función Objetivo del *PPE* en PDDL.

```

(:init
  (free student1)
  (= (total-credits-subject-gain student1) 0)
  (= (available-credits student1) 48)

  (available-subject Material1 student1)
  (= (credits-subject Material1) 9)
  (= (mingrade Material1) 70)
  (not-approved Material1 student1)
  (not-done-LA LA1 Material1 student1)
  (not-done-LA LA2 Material1 student1)
  (not-done-LA LA3 Material1 student1)
  (not-done-LA LA4 Material1 student1)
  (not-done-LA LA5 Material1 student1)
  (= (score Subtema1 student1) 0)
  (not-done-LA LA6 Material1 student1)
  (not-done-LA LA7 Material1 student1)
  (not-done-LA LA8 Material1 student1)
  (not-done-LA LA9 Material1 student1)
  (not-done-LA LA10 Material1 student1)
  (= (score Subtema2 student1) 0)

  (= (quantity-resource rec0)100000)
  (= (quantity-resource rec1) 20)
  (= (quantity-resource rec2) 30)
  (= (quantity-resource rec3) 40)

  (= (valueLA LA1) 15)
  (= (valueLA LA2) 22)
  (= (valueLA LA3) 21)
  (= (valueLA LA4) 15)
  (= (valueLA LA5) 27)
  (= (valueLA LA6) 19)
  (= (valueLA LA7) 22)
  (= (valueLA LA8) 23)
)

```

Figura 2.3: Condiciones iniciales de variables para el *PPE* en PDDL.

Y un dominio donde esta la descripción del problema, figura 2.4, una descripción de dominio se puede utilizar para muchos problemas diferentes, generando muchos problemas de planificación para un mismo dominio.

```

(define (domain degree)
  (:requirements :durative-actions :typing :fluents :equality)
  (:types student resource - object
    subject Theme subtheme LA - LO)
  (:constants
    Materia1 - subject

    Tema1 - Theme

    Subtema1
    Subtema2 - subtheme

    LA1
    LA2
    LA3
    LA4
    LA5
    LA6
    LA7
    LA8
    LA9
    LA10 - LA

    rec0
    rec1
    rec2
    rec3 - resource
  )

```

Figura 2.4: Parte del dominio de un PPE en PDDL.

La parametrización de las acciones depende del uso de variables que representan términos de las instancias del problema. Las pre y post condiciones de las acciones son expresadas como proposiciones lógicas construidas de predicados, conectores lógicos y argumentos, figura 2.5 (instancias de objetos del problema).

```

(:durative-action CHOOSE-LA-nothasreqs
:parameters (?s - student ?oa - LA ?subt - subtheme ?t - Theme ?subj - subject ?eq - resource)
:duration (= ?duration (DurationLA ?oa))
:condition (and
  (at start (free ?s))
  (at start (enrollment ?s ?subj))
  (at start (not-done-LA ?oa ?subj ?s))
  (at start (isPartOfSubtheme ?oa ?subt))
  (at start (isPartOfTheme ?subt ?t))
  (at start (isPartOfSubject ?t ?subj))
  (at start (KindResourceLO ?oa ?eq))
  (at start (> (quantity-resource ?eq) 0))
  (at start (not-has-reqs ?oa))
  (at start (> (maxgrade-subtheme ?subt)(valueLA ?oa)))
)
:effect (and
  (at start (not(free ?s)))
  (at start (decrease (quantity-resource ?eq) 1))
  (at end (increase (quantity-resource ?eq) 1))
  (at end (not (not-done-LA ?oa ?subj ?s)))
  (at end (increase (score ?subt ?s) (valueLA ?oa)))
  (at end (free ?s))
  (at end (decrease (maxgrade-subtheme ?subt)(valueLA ?oa)))
  (at end (done ?oa))
)
)

```

Figura 2.5: Instanciación de objetos de un PPE en PDDL.

Un ejemplo completo de un problema de planificación educativa codificado en PDDL se puede encontrar en el apéndice A .

## 2.3 OPTIC

*OPTIC (Optimizing Preferences and Timedependent Costs)* [2] es un planificador temporal con encadenamiento hacia adelante, propio de un planificador basado en estados que incorpora ideas de planificación parcial dándole así un poco de flexibilidad, fue desarrollado por *Andrew et. al.* en la 22<sup>nd</sup> conferencia sobre planificación automatizada y secuenciación, se desarrolló a partir de otro planificador temporal llamado *POPF* [4].

### 2.3.1 INTRODUCCIÓN A OPTIC

Mucho tiempo se ha trabajado dentro de la investigación de planificación temporal, sobre la disminución del *makespan* asumiendo que el tiempo está estrechamente relacionado con la utilidad del plan.

El *makespan* lo podemos definir como el tiempo transcurrido desde el inicio del proceso hasta el final.

Sin embargo, el *makespan* muchas veces no se alinea con los objetivos globales del problema. Un ejemplo de esto es el problema de logística publicado por *Andrew Coles et. al.* en el artículo *Temporal Planning with Preferences and Time-Dependent Continuous Costs*.

Se tiene que entregar un cargamento de moras, naranjas y manzanas en las locaciones  $B, O$  y  $A$  respectivamente desde el punto  $P$ , cada fruta tiene diferente tiempo de vida después de ser recolectadas. Las moras duran 10 días, las naranjas duran 15 días y las manzanas duran 20 días. El tiempos para ir del punto  $P$  al  $B$  es de 6 días, entre  $P$  y  $A$  es de 7 días, entre  $B$  y  $O$  es de 3 días y entre  $A$  y  $B$  es de 5 días. Para realizar todas las entregas el plan más corto tiene una duración de 15 días que sería realizar el siguiente recorrido ( $A \Rightarrow B \Rightarrow O$ ), si la entrega se realiza en este orden, las moras y las naranjas no soportarían el viaje. Este es un claro ejemplo de como no siempre el *makespan* más pequeño se alinea con los objetivos principales del problema.

La diferencia de *OPTIC* con otros planificadores temporales como *CRIKEY*, *COLIN* y *POPF* [1, 3, 4] es la manera en como maneja las restricciones temporales, *OPTIC* incorpora características de *PDDL3* permitiéndole manejar restricciones temporales continuas, por ejemplo, cuando en un determinado problema se sobrepasa el tiempo ideal para cierta actividad se incurre en una penalización establecida previamente, en *OPTIC* al sobrepasar este *deadline* se puede penalizar con zero para una cierta cantidad de tiempo y después incurrir en una penalización que se aumenta progresivamente.

Además del manejo de restricciones temporales continuas, algunas de las principales características de *OPTIC* son:

- La generación de estados sucesores durante la búsqueda en el espacio de estados es muy rápida ya que solo considera acciones soportadas completamente.

Si ejemplificamos esto en un contexto del PPE, supongamos que existe la acción *add act3*, pero en planificación temporal, las acciones se dividen en dos, una acción de inicio (*start add act3*) y una de fin (*end add act3*). Por lo que, la acción *end add act3* no puede aparecer si no ha terminado la acción *start add act3*, por lo que no se considera como posible acción a aplicar.

- Sólo agrega acciones que sus precondiciones fueron cumplidas, por lo que las nuevas acciones no amenazan las precondiciones de acciones anteriores.

### 2.3.2 REPRESENTACIÓN DE PLANES MEDIANTE GRAFOS

El planificador *OPTIC* hace uso de un tipo de grafo en particular conocido como *TRPG* (*Temporal Relaxed Plan Graph*) para guiar su búsqueda en el espacio de estados, la cual hace uso de un tipo de grafo que es una adaptación del *RPG* introducido por *Hoffman* [12], donde se pueden representar la simulación de la ejecución de las acciones de un plan. Este grafo fue introducido en *Sapa: A Multi-objective Metric Temporal Planner* por *Subbarao Kambhampati* y *Minh B. Do* en el 2003 y es uno de las estructuras más utilizadas en planificación temporal.

#### TRPG

Para la representación de planes temporales en *PDDL 2.1* se dividió las acciones temporales en dos acciones conocidas como *snap-actions*, una que marca el inicio de una acción y una el fin, figura 2.6. En *OPTIC* las acciones de fin siempre tienen que estar soportadas por una de inicio.

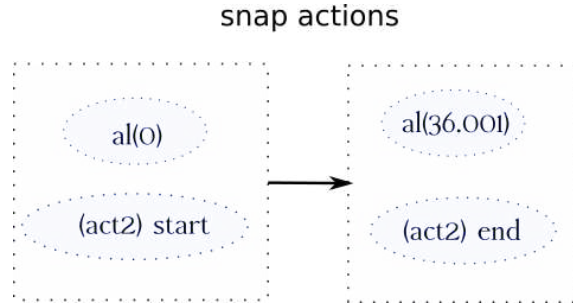


Figura 2.6: Ejemplo de una Acción Temporal dividida en dos, una de inicio y una de fin. La etiqueta  $al$  hace referencia al *makespan* acumulado.

El grafo consiste en intercalar capas de hechos y acciones, las cuales van surgiendo de la simulación de la aplicación de las acciones hasta una capa donde todos los objetivos estén satisfechos, con la finalidad de obtener un valor aproximado del tiempo restante para satisfacer todos los objetivos.

Un ejemplo de este grafo es el que se observa en la figura 2.7, los recuadros en gris representan las capas de hechos y los recuadros blancos las capas de acciones, ambas capas se van intercalando durante la expansión de grafo.

Las capas de acción y de hechos tienen asociado un valor numérico que corresponde con el tiempo en el que se puede alcanzar esa capa. el orden de las acciones se establece de tal manera que finalicen lo antes posible.

En este ejemplo podemos observar que las capas de hechos contienen aquellos hechos soportados por las capas de acciones anteriores, por ejemplo, la capa de hechos con un tiempo de  $fl(30.002)$  *act5 terminada*, no podría haber aparecido si las acciones de  $(est1 \ act5 \ sub1 \ mat1 \ rec3) \text{ start}$  y  $(est1 \ act5 \ sub1 \ mat1 \ rec3) \text{ end}$  no hubieran aparecido antes en ese orden, puesto que una acción de finalización no puede aparecer sin una acción de inicio que la soporte.

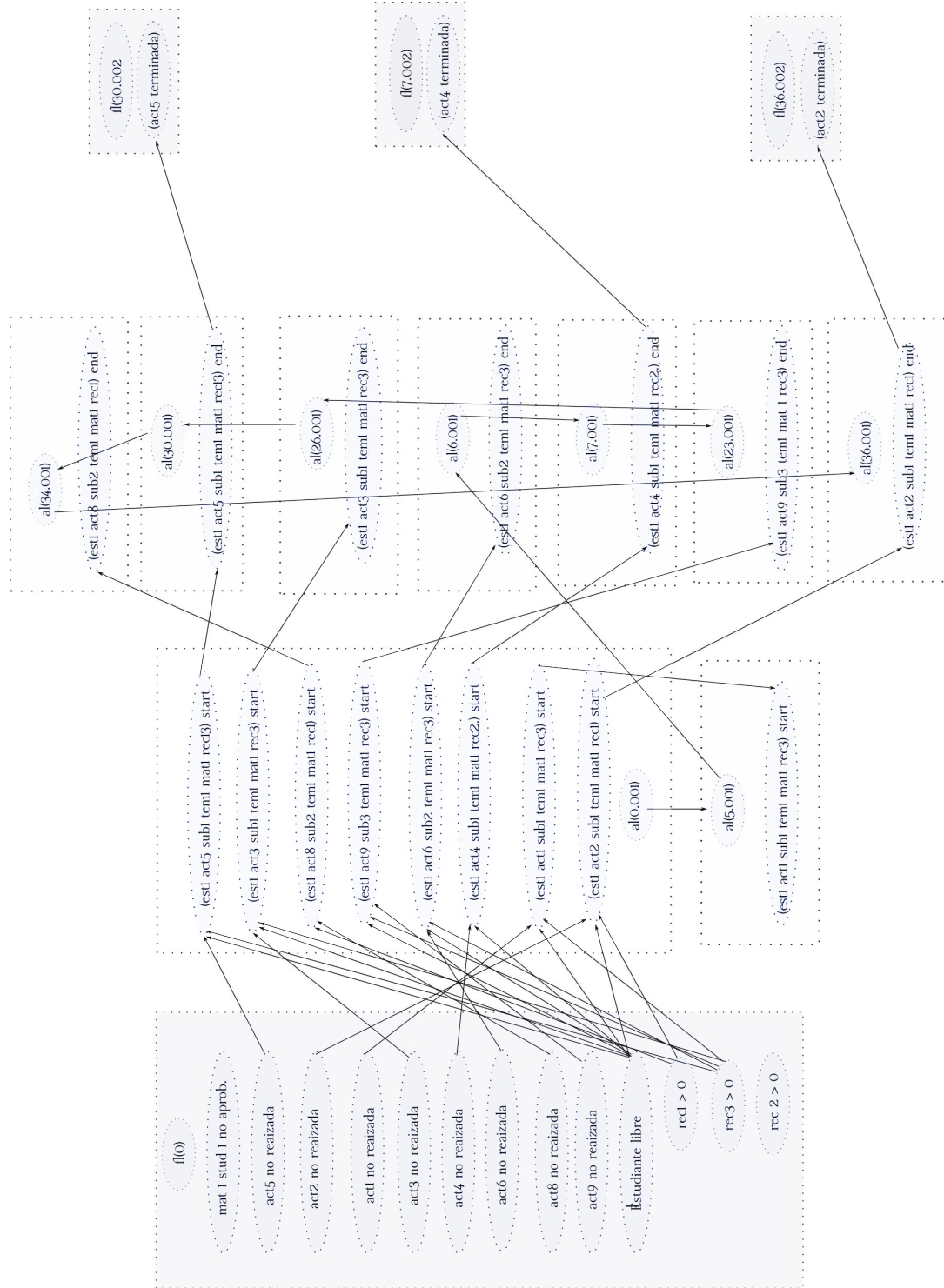


Figura 2.7: Parte de un grafo TRPG de un Problema de Planificación Educativa.

El algoritmo 2 describe como se genera un *TRPG*.

**Algoritmo 2** Algoritmo para la generación de un TRPG.

---

```

1: procedure TRPG
2:   Data:  $S = \langle F, E, T \rangle$  – estado a evaluar
3:   Resultado:  $R = \langle f_{0...n}, a_{0...n} \rangle$  – Un RPG
4:    $f_0 \leftarrow F$ 
5:    $t \leftarrow 0$ 
6:   for  $A_{-1}$  do
7:     if  $(e \in E \mid e.op = A) = 0$  then
8:        $earliest(A) \leftarrow \infty$ 
9:     else
10:       $earliest(A) \leftarrow 0$ 
11:    end if
12:  end for
13:  while  $t < \infty$  do
14:     $f_{t+e} \leftarrow f_t$ 
15:     $fa_t \leftarrow \{ A_{-1} \mid pre(A_{-1}) \subseteq f_t \wedge earliest(A) \leq t \}$ 
16:    for new  $A_{-1} \in a_t$  do
17:       $f_{t+e} \leftarrow f_{t+e} \cup eff^+(A_{-1})$ 
18:    end for
19:     $a_t \leftarrow a_t \cup \{ A_{-1} \mid pre(A_{-1}) \subseteq f_t \}$ 
20:    for new  $A_{+1} \in a_t$  do
21:       $f_{t+e} \leftarrow f_{t+e} \cup eff^+(A_{+1})$ 
22:       $earliest(A) = \min[earliest(A), t + lb(A)]$ 
23:    end for
24:    if  $f_t \subset f_{t+e}$  then
25:       $t \leftarrow t + \epsilon$ 
26:    else
27:       $ep = \{ earliest(A) > t \mid pre(A_{-1}) \subseteq f_t \}$ 
28:      if  $ep \neq \emptyset$  then
29:         $t \leftarrow \min[ep]$ 
30:      else
31:        break
32:      end if
33:    end if
34:  end while return  $R = \langle f_{0...n}, a_{0...n} \rangle$ 
end procedure

```

---

La suma de los pasos faltantes para satisfacer los objetivos es una heurística muy utilizada en planificación, en el caso de los planificadores temporales como *OPTIC* no sería los números de pasos si no la duración de las acciones en realidad el valor que se toma en cuenta.

El planificador *OPTIC* hace uso de dos tipos de estructuras para resolver los problemas, un grafo de estados donde cada arco corresponde con una acción



posible de aplicar y cada nodo corresponde con un estado resultante de aplicar ciertas acciones, dentro de cada nodo existe un *TRPG* 2.8 que sería la simulación de la ejecución de las acciones permitidas en el estado actual.

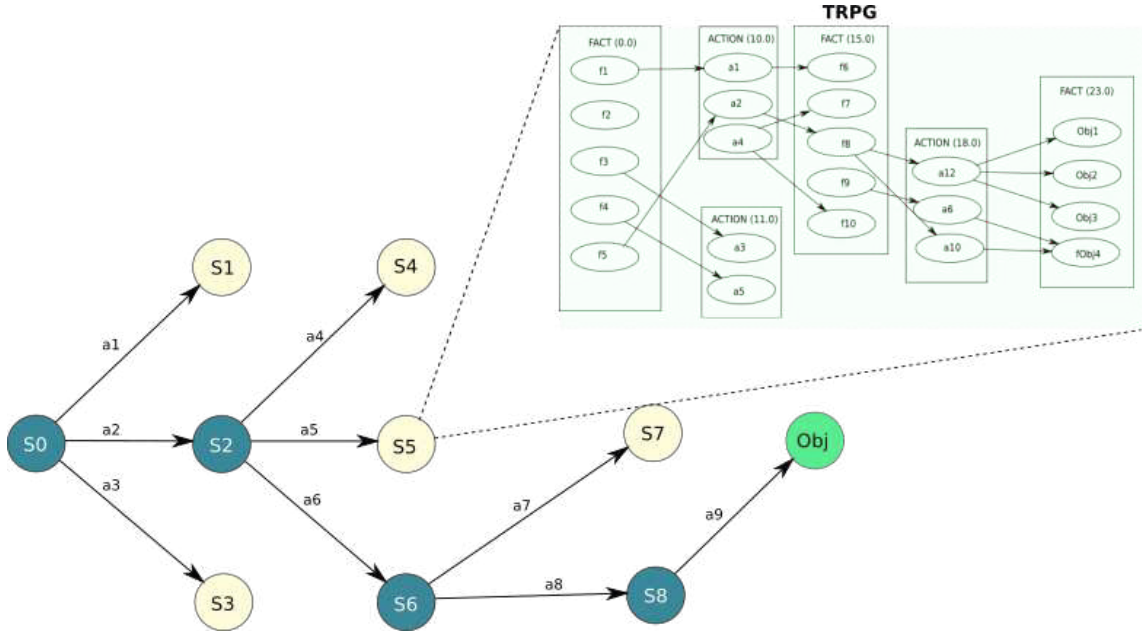


Figura 2.8: Representación gráfica de un Grafo de Estados de *OPTIC*.

### 2.3.3 BÚSQUEDAS MÁS COMUNES

Durante la expansión de los nodos en el grafo de estados, se hace uso de búsquedas primero el mejor mediante una función heurística  $h(s)$  que calcula el menor costo estimado de alcanzar el nodo final desde el nodo actual.

#### ASCENSO DE COLINAS

*OPTIC* hace uso de el método de Ascenso de Colinas para la búsqueda dentro del grafo de estados. Ascenso de Colinas es una heurística muy utilizada en muchos planificadores como estrategia de búsqueda, se introdujo en *Fast Forward* [14] por *Hoffmann and Nebel* en el 2001.

La búsqueda de ascenso de colinas es una búsqueda simple que se mueve ha-

cia el primer mejor vecino en lugar de hacia el mejor vecino utilizando una función heurística para calcular la distancia aproximada desde el estado actual a un estado objetivo, la búsqueda la hace en la siguiente capa de estados como un *breadth first search*, esta selección ciega del siguiente nodo puede disminuir la eficiencia del algoritmo, especialmente en grafos con muchos caminos.

---

**Algoritmo 3** Ascenso de Colinas.
 

---

```

1: procedure BUSQUEDAASCENSOCOLINAS
2:   Inicializa el plan actual a un plan vacío <>
3:    $S := I$ 
4:   while  $h(S) \neq 0$  do
5:     Realiza un BFS para un estado  $S'$  con  $h(S') < h(S)$ 
6:     if Si no se encuentra un estado then
7:       return Fail
8:     end if
9:     Agrega la acción del camino hacia  $S'$  al fin del plan actual
10:     $S := S'$ 
11:  end while
12: end procedure

```

---

El algoritmo lo que trata de hacer es *minimizar* (*maximizar*) una función  $h(s)$ , donde  $s$  representa un estado. Los estados son típicamente representados mediante grafos como nodos y las acciones como aristas.  $HC$  traza la ruta en el grafo de búsqueda nodo por nodo decreciendo (incrementando) el valor de  $h$  hasta un mínimo (máximo) local.

Generalmente el algoritmo de ascenso de colinas es muy rápido ya que es fácil salir de un estado malo pero también puede llegar a situaciones donde le es difícil avanzar como las siguientes situaciones, figura 2.9:

- Máximo Local: Encontrara un máximo local es llegar a un pico que es el punto más alto que cada uno de sus vecinos pero más bajo que el máximo global.

- Mesetas: Es un área plana del espacio de estados del que no existe ninguna salida ascendente, este tipo de situaciones se pueden resolver realizando un número de iteraciones para tratar de llegar a un estado en el que se es posible mejorar o empeorar en cuyo caso terminaría la búsqueda.
- Risco u Hombro: Es un área plana del espacio de estados similar a una meseta pero que después de un cierto número de movimientos en una dirección en particular se puede salir y avanzar a mejores estados.

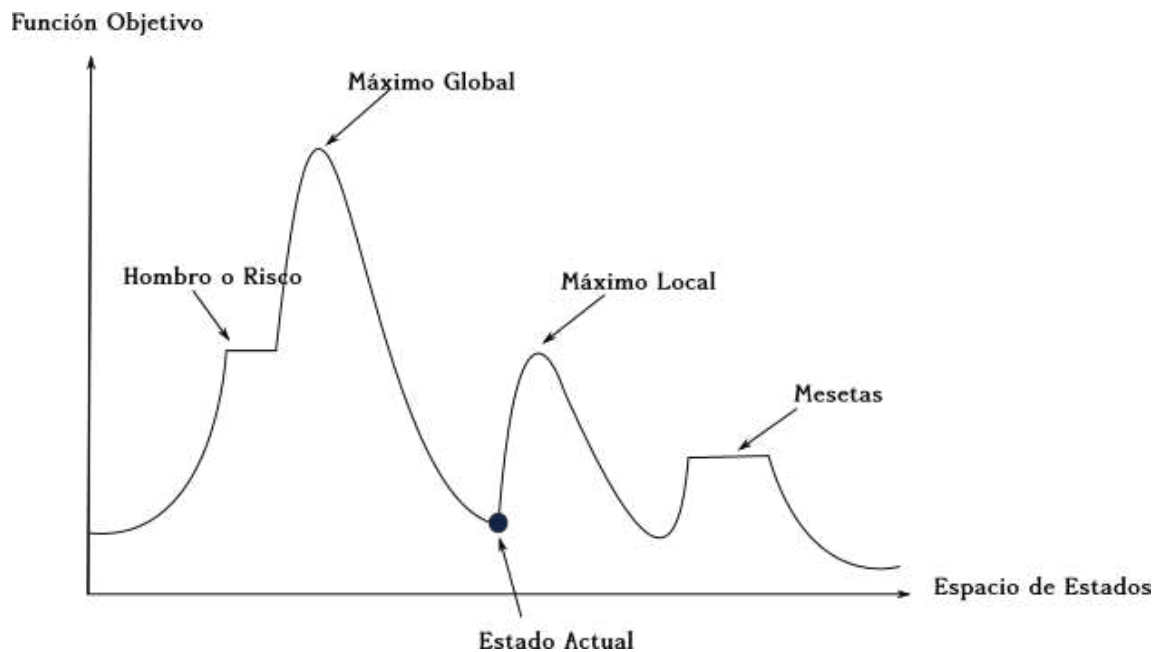


Figura 2.9: Imagen para Algoritmo Ascenso de Colinas

Una variante del algoritmo es el EHC(Enforced Hill Climbing), que es una combinación de un Hill Climbing y un Breadth First Search.

**Algoritmo 4** Enforced Hill Climbing

---

```

1: procedure EHC
2:   Entrada:  $P \leftarrow \langle \text{estado } s, \text{ acciones } A, \text{ objetivos } G \rangle$ 
3:    $h_s \leftarrow h(s, A, G)$ 
4:    $Lista \leftarrow \text{sucesores}(s)$ 
5:   while  $Lista \neq 0$  do
6:      $\text{sig\_estado} \leftarrow Lista.pop()$ 
7:      $h_{\text{sig\_estado}} \leftarrow h(\text{sig\_estado}, A, G)$ 
8:     if  $h_{\text{sig\_estado}} = 0$  then
9:       return Se encontró un plan
10:    else if  $h_{\text{sig\_estado}} < h_s$  then
11:       $EHC(\text{sig\_estado}, A, G)$ 
12:    else  $Lista \leftarrow Lista + \text{sucesor}(\text{sig\_estado})$ 
13:    end if return Se alcanzó un deadend
14:  end while
15: end procedure

```

---

Donde sin la línea 12 del algoritmo 4 se convierte en un simple *HC*, este reemplaza el estado actual por el primer mejor estado encontrado. Ahora sin la línea 10 el algoritmo se transforma en un *BFS*.

Una de las limitaciones del *EHC* es que puede quedar atrapado en situaciones con mínimos locales similar al *HC*, en estos casos las tareas de planificación contienen estados para los cuales existen objetivos que no se pueden satisfacer, en esos casos recurren a algoritmos de primero el mejor como un *BFS*.

### 2.3.4 RESOLVIENDO UN PROBLEMA CON OPTIC

*OPTIC* extiende las capacidad de el secuenciador de *POPF*, la heurística y la estrategia de búsqueda. Para realizar la búsqueda es necesario asignarle un valor  $f(s)$  a cada nodo de tal manera que se pueda seleccionar el primero mejor encontrado. El

valor de cada nodo esta dado por la ecuación 2.1.

$$f(s) = g(s) + h_a(s). \quad (2.1)$$

Donde  $g(s)$  es el costo real de alcanzar el nodo  $s$  y  $h(s)$  es un valor heurístico obtenido por el *TRPG*.

Durante la búsqueda de soluciones se usa también el valor de  $h(s)$  estimado para podar estados y agilizar la búsqueda. Esto ocurre si la suma de un estado alcanzado  $s$  con un costo  $g(s)$ , con un valor heurístico  $h(s)$  sobrepasa un valor  $c$ , si  $g(s) + h_a(s) \geq c$  donde  $c$  es una cota superior en el costo se puede podar el estado.

La búsqueda es eficiente ya que  $h(s)$  nunca sobrestima el costo de llegar a un objetivo y por naturaleza es una búsqueda optimista ya que supone que el costo de resolver el problema es menor que el que es en realidad. Y como  $g(s)$  es el costo original de alcanzar  $s$  tenemos que  $f(s)$  nunca sobrestima el costo de una solución a través de  $s$ .

Una vez encontrada una solución *OPTIC* puede mejorar esa solución usando como cota superior el valor alcanzado.

Para clarificar la manera en como *OPTIC* realiza la búsqueda de una solución tomemos como ejemplo el problema representado en la figura 1.1.

Para el problema de ejemplo de la figura 1.1, *OPTIC* encuentra la siguiente solución mostrada en la tabla 2.1.

Resultados del plan de OPTIC	
Dato	Valor
Métrica	162.011
Estados Evaluados	24
Estados Podados	0
Tiempo de Cómputo	0.11 Segundos

Tabla 2.1: Primer solución encontrada por el planificador OPTIC.

La secuencia de actividades de la solución encontrada se puede observar en la tabla 2.2.

Secuencia de Acciones	
Métrica	Acción
0.000	(enroll-subject material1 student1)
0.001	(choose-la-nothasreqs student1 la9 subtema2 tema1 material1 rec3)
23.002	(choose-la-nothasreqs student1 la6 subtema2 tema1 material1 rec3)
29.003	(choose-la-nothasreqs student1 la5 subtema1 tema1 material1 rec3)
59.004	(choose-la-hasreqsla student1 la7 subtema2 tema1 material1 rec1 la5)
90.005	(choose-la-nothasreqs student1 la1 subtema1 tema1 material1 rec3)
95.006	(choose-la-nothasreqs student1 la4 subtema1 tema1 material1 rec2)
102.007	(choose-la-nothasreqs student1 la3 subtema1 tema1 material1 rec3)
128.008	(choose-la-nothasreqs student1 la8 subtema2 tema1 material1 rec1)
162.009	(pass-theme-tema1 material1 student1)
162.010	(pass-material1 student1)
162.011	(take-subject-pass student1 material1)

Tabla 2.2: Primer solución encontrada por el planificador OPTIC.

Como se menciona anteriormente *OPTIC* es un planificador con encadenamiento hacia delante que hace uso de un grafo *TRPG* que es un *RPG* con acciones

temporales, este se expande hacia adelante hasta encontrar una capa donde todos los objetivos estén satisfechos, en este caso que se apruebe la *materia1* con la acción (*take-subject-pass student1 materia1*) que nos lleva al estado (*pass-degree materia1 student1*).

Una vez alcanzada esa capa se realiza un encadenamiento hacia atras para encontrar el mejor camino hacia el punto de partida, esto servirá para poder obtener el valor  $h(s)$  el cual es necesario para la búsqueda con el algoritmo *EHC* que necesita el valor de cada estado.

Este proceso se realiza hasta que el valor heurístico sea 0 o no se pueda encontrar un camino para satisfacer todos los objetivos.

Cuando *OPTIC* encuentra una solución, se prosigue con una fase de mejora, la cual en base a la solución inicial encontrada toma la métrica de calidad como cota superior ayudando así a cortar ramas con un algoritmo  $WA^*$  y alcanzar una mejor solución.

## CAPÍTULO 3

# METODOLOGÍA

---

Una desventaja de los planificadores es el crecimiento del espacio de búsqueda en problemas grandes, por lo que aprovechando esa desventaja se pretende crear soluciones mediante un algoritmo constructivo las cuales servirán para reducir el espacio de búsqueda y tratar de encontrar una mejor solución.

Para lograr esto el planificador OPTIC cuenta ya con un procedimiento que reduce su espacio a partir de la solución anterior encontrada, el cual utiliza el valor de esas solución para podar ramas, similar a *branch and bound* [5] .

**Definición 3.0.1. Branch and Bound.** El algoritmo busca dentro de la región factible un valor óptimo mediante la eliminación de regiones donde muy probablemente no exista esa solución. Este algoritmo utiliza cotas para poder eliminar aquellas ramas que no conducen a una mejor solución, si la cota es mejor que el valor de un nodo dado, esa rama se elimina ya que no nos conduciría a una mejor solución.

Una manera de obtener buenas soluciones iniciales de forma rápida para el planificador es utilizar un constructivo de soluciones aleatorias (al que nos referiremos de ahora en adelante como *CSA*), el cual respeta las restricciones del problema y genera soluciones que servirán como cotas, esto para evitar el tiempo de computo al buscar una solución inicial por el propio planificador.



### 3.1 DESCRIPCIÓN DEL CONSTRUCTIVO

Para la obtención de soluciones de calidad se generó el siguiente algoritmo constructivo 5. Su funcionamiento aprovecha la rapidez de generar soluciones aleatorias y por lo mismo la capacidad de generar un gran número de soluciones, esto debido a que cuando se agrega una acción a una solución no se cuestiona si es la mejor o no solo si se puede agregar, al hacer esto el proceso se vuelve más ágil.

Lo primero que se hace es generar listas para ir almacenando las actividades seleccionadas (*seleccion*), los puntos acumulados por subtema (*puntos*) y las actividades eliminadas (*eliminadas*); una vez generadas estas estructuras se verifica si existen actividades obligatorias, si es así, se agregan a la lista *seleccion* y se suman los puntos correspondientes a cada subtema, en caso de que una actividad obligatoria tenga prerequisites, también se agregan los prerequisites esto para que tenga congruencia con las restricciones del problema.

Una vez agregadas las actividades obligatorias se procede a la selección de actividades, se seleccionaran actividades mientras aun existan subtemas en la lista *puntos* con un valor menor a la métrica utilizada por el problema, en este caso todos los problemas buscan obtener mayor o igual a 70 y menor o igual a 100.

Las actividades se seleccionan aleatoriamente, se elige un número entre 0 y la cantidad de actividades restantes, si el subtema de la actividad seleccionada ya tiene más de 70 puntos o es mayor a la métrica del problema, esa actividad se elimina sin sumar puntos y se selecciona otro número, otro caso sería si al seleccionar una actividad los puntos que suma sobrepasan el valor superior que en este caso es 100, esa actividad se descarta y se toma otra.

Una vez encontrada una actividad cuyo subtema aun tenga menos a 70 se procede a verificar si cuenta o no con restricciones de habilitamiento, puede tener una o dos restricciones.

Si tiene restricciones, se verifica que no hayan sido agregadas ya a la lista

*seleccion*, si ya se agregaron solo se agrega la actividad, si aun no han sido agregadas se agregan sus requisitos y luego la actividad en ese orden, también se suman los puntos correspondientes a cada uno.

El proceso sigue hasta que todos los subtemas tengan más de 70 puntos o 70 y se devuelve la secuencia de actividades almacenadas en la lista *seleccion*.

---

**Algoritmo 5** Constructivo
 

---

**Require:**

```

1: i: Iterador de cuantas soluciones generara.
2: procedure CSA(i)
3:   if act.obligatorias != null then
4:     for o in act.obligatorias do
5:       plan = plan + o
6:       o.subtema = o.subtema + o.valor      ▷ Se acumula el valor de la actividad al subtema.
7:     end for
8:   end if
9:
10:  while Existan subtemas <70 do
11:    a ← Selecciona una actividad de manera aleatoria.
12:    while a.req != null do
13:      if a.req no esta en el plan then plan = plan + a.req
14:      plan = plan + a
15:    end if
16:  end while
17: end while
18: return plan
19: end procedure

```

---

El diagrama de flujo de la figura 3.1 muestra como sería el flujo de datos dentro del constructivo y como se van incorporando actividades dentro de una solución.

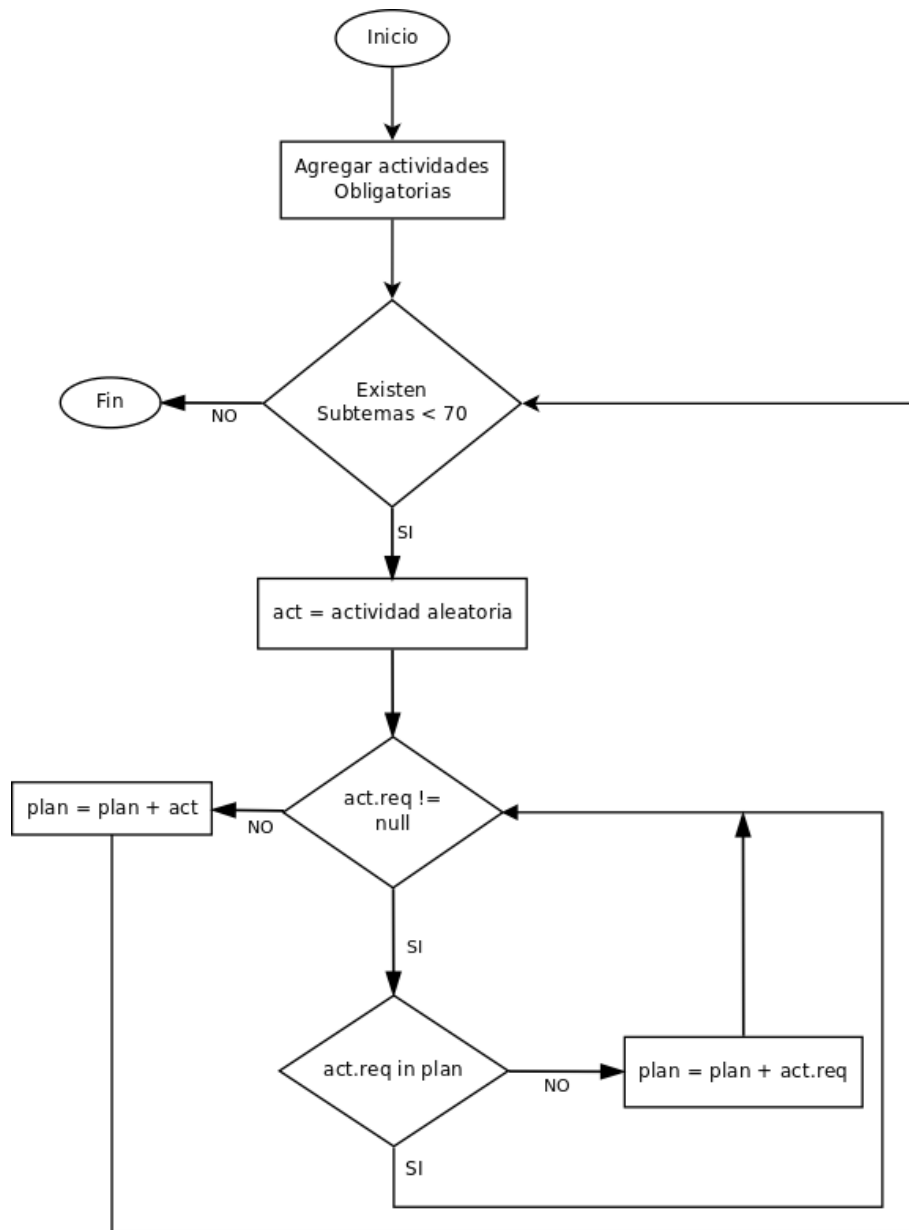


Figura 3.1: Diagrama de Flujo del CSA.

A cada problema se le extrajo la información necesaria para resolverlo con el CSA y se guardó en un archivo de texto con un formato similar a la tabla 3.1 de manera que fuese fácil leerlo.

Formato de las instancias										
Materias	1									
Temas	1									
Subtemas	2									
Actividades	10									
Actividades con Requisitos	3									
Duración	23	15	21	23	18	16	23	21	19	21
Valor	39	19	34	29	9	5	25	33	13	36
Recurso	3	3	3	1	3	3	2	2	1	2
Actividades del subtema 1	1	2	3	4	5					
Actividades del subtema 2	6	7	8	9	10					
Subtemas del Tema 1	1	2								
Temas de la Materia 1	1									
Requisito 1	10	4								
Requisito 2	9	1								
Requisito 3	9	5								
Actividades obligatorias	0									

Tabla 3.1: Formato de los datos extraídos de un problema de ejemplo.

## 3.2 OBTENCIÓN DE RESULTADOS CON OPTIC

Para obtener las soluciones mediante el planificador se le impuso un tiempo límite a la búsqueda ya que como se explico en la sección anterior, al incrementar el número de materias y actividades en el problema el espacio de búsqueda crece exponencialmente por lo que el proceso de búsqueda de una solución suele ser muy tardado, los tiempos límite que se usaron fueron 5,10,20 y 60 minutos siendo este ultimo el tiempo máximo para la búsqueda de una solución o tratar de mejorar la ya encontrada y donde se lograron obtener un mayor número de soluciones.

Para las instancias con una sola materia 20 minutos fueron suficiente para encontrar una solución inicial, pero para las instancias con más de una materia fue necesario incrementar hasta una hora ya que en muchos casos no lograba encontrar una solución inicial.

### 3.3 OBTENCIÓN DE EL VALOR ÓPTIMO

Para la obtención del valor óptimo se uso *GAMS(General Algebraic Model System)* que es sistema de modelado para programación matemática y de optimización integrado con un conjunto de solvers.

Para obtener una solución óptima en el trabajo *Generación de trayectorias de aprendizaje a través de modelos educativos de planificación* desarrollado por Maya P. et al. [9] generaron el siguiente modelo matemático, el cual relajó la obtención de una secuencia como solución, obteniendo un modelo de selección de actividades, el cual nos permite saber que actividades deben de ser incluidas en la secuencia:

Función Objetivo

$$\min \sum_{i=1}^{\eta} \sum_{j=1}^{\alpha} d_{ij} x_{ij}. \quad (3.1)$$

Para la función objetivo 3.1 se busca minimizar el tiempo que se necesita para aprobar una o varias materias según un conjunto de actividades seleccionadas. Esta función esta sujeta a algunas restricciones, como los es un puntaje mínimo  $k_{min}$  3.2 y uno máximo  $k_{max}$ , restricción 3.3.

$$\sum_{i=1}^{\eta} u_{ij} x_{ij} \geq Kmin \quad j = 1, 2, \dots, \alpha. \quad (3.2)$$

$$\sum_{i=1}^{\eta} u_{ij} x_{ij} \leq Kmax \quad j = 1, 2, \dots, \alpha. \quad (3.3)$$

Para saber si una actividad se considera o no dentro de un plan se utiliza una variable binaria  $x_{ij}$  que valdría 1 si se considera dentro del plan y 0 si no 3.4.

$$x_{ij} \in 0, 1 \quad i = 1, 2, \dots, n; j = 1, 2, \dots, \alpha. \quad (3.4)$$

Como se observa en la ecuación 3.5 el puntaje alcanzado para el subtema  $j$  se refleja mediante  $y_j$  que es la suma del valor de la actividad  $i$  del subtema  $j$  siempre y cuando  $x_i$  sea igual a 1.

$$y_j = \sum_{i=1}^{\eta} u_{ij} x_{ij} \quad j = 1, 2, \dots, \alpha. \quad (3.5)$$

$$y_j \geq 0 \quad j = 1, 2, \dots, \alpha. \quad (3.6)$$

$$x_{i'j'} \leq \omega_{ii'} x_{ij} \quad i, i' = 1, 2, \dots, n; j, j' = 1, 2, \dots, \alpha. \quad (3.7)$$

También se consideran actividades obligatorias para algunos subtemas 3.8, por lo que para esas actividades  $x_{ij}$  siempre sera 1.

$$x_{ij} = 1 \quad \forall m_{ij} \in M. \quad (3.8)$$

### 3.4 EXTENSIÓN DEL MODELO MATEMÁTICO DEL PPE

En la sección 3.3 se describió el problema de planificación educativa y las restricciones que se consideraron en el trabajo realizado por Maya P. et al. [9]. En esta sección se describen las restricciones con las cuales se extiende el modelo matemático presentado y que representan algunas características subjetivas del estudiante, que afectan en gran medida la manera en como se seleccionan las actividades que conforman un plan y es de gran importancia para la investigación de educación a distancia.

Para este trabajo solamente se consideraran tres características que afectarían el desempeño de un plan en particular. Las características son: Estrés, Calidad del material y Capacidad intelectual.

También se consideraron restricciones para no solo obtener que conjunto satisface mis objetivos, si no, también obtener en que secuencia deben de ir las actividades seleccionadas. Además, se agregó una nueva característica a las actividades, como lo es la complejidad de una actividad en particular, la cual se ve afectada por los factores mencionados anteriormente, el estrés, la calidad del material y la posición en la que se ejecutara dicha actividad.

### 3.4.1 RESTRICCIONES PARA OBTENER UNA SECUENCIA

Para lograr obtener una secuencia a partir del modelo matemático anterior, fue necesario agregar un sub índice mas en la función objetivo y en algunas restricciones que se veían afectadas por la posición de las actividades.

Al agregar este nuevo sub índice la función objetivo queda como la que se observa en la ecuación 3.9.

#### Definición de Variables

$d_{ij}$  .- Duración de la actividad  $i$  del subtema  $j$ .

$x_{ijq}$  .- Variable binaria para denotar que se agrego la actividad  $i$  del subtema  $j$  en la posición  $q$ .

#### Función Objetivo

$$\text{mín} \sum_{i=1}^{\eta} \sum_{j=1}^{\alpha} \sum_{q=1}^{\beta} d_{ij} x_{ijq}. \quad (3.9)$$

La restricción de la ecuación 3.10 nos permite asegurar que no existan dos actividades se ejecuten al mismo tiempo, es decir, no existen actividades que tengan una misma posición en la secuencia.

$$\sum_{i=1}^{\eta} \sum_{j=1}^{\alpha} x_{ijk} \leq 1 \quad k = 1, 2, \dots, \beta. \quad (3.10)$$

### 3.4.2 MODELACIÓN DEL ESTRÉS

Otra restricción a considerar, es el estrés generado por la actividad  $i$  del subtema  $j$  en la posición  $q$ , el cual definiremos como  $f_{ijq}$ , el cual se verá afectado por la diferencia entre el puntaje acumulado ( $y_j$ ) del subtema  $j$  y la calificación mínima ( $K_{min}$ ), es decir cuanto falta para alcanzar el puntaje mínimo en ese subtema.

El estrés puede ser modelado de distintas formas, dependiendo de cuantos factores se consideren, una forma sería como se observa en la ecuación 3.11, donde el estrés se ve afectado por la diferencia entre el puntaje acumulado y el puntaje mínimo ( $\frac{1}{K_{min}-y_j}$ ) y un tiempo restante para alcanzar el puntaje mínimo ( $\frac{1}{\Delta_T-\Delta_A}$ ) donde  $\Delta_T$  es una constante que indica el tiempo disponible para satisfacer un puntaje mínimo y  $\Delta_A$  el tiempo utilizado hasta el momento.

Sin embargo, esta función no representa el estrés independiente del tiempo, es decir,  $\frac{1}{K_{min}-y_j}$  no disminuye el estrés conforme la diferencia entre el puntaje mínimo ( $K_{min}$ ) y el puntaje acumulado ( $y_j$ ) se hace más pequeña.

#### Definición de Variables

$\Delta_T$  .- Tiempo disponible para satisfacer un puntaje mínimo en todos los subtemas.

$\Delta_A$  .- Tiempo utilizado hasta el momento.

$K_{min}$  .- Puntaje mínimo necesario.

$y_j$  .- Puntaje acumulado del subtema  $j$ .

$$f_{ijq} = 1 + \left( \frac{1}{K_{min} - y_j} \right) * \left( \frac{1}{\Delta_T - \Delta_A} \right) \quad (3.11)$$

Otra forma de modelar el estrés es la observada en la ecuación 3.12 la cual considera también un factor de tiempo mediante  $(\Delta_T - \Delta_A)^{\frac{1}{\Delta_A}}$ , pero la forma en como se evalúa el estrés  $(K_{min} - y_j)^{\frac{1}{y_j}}$  no es completamente dependiente de ese factor



y el tiempo solo funciona como incremento del estrés, a menor tiempo para cumplir con una calificación mínima, mayor estrés, como sucede en la vida real. Por lo que, si quitamos el factor de tiempo de la ecuación, la función  $f_{ijq}$  sigue representando el estres, como se observa en la ecuación 3.13.

**Considerando el tiempo:**

$$f_{ijq} = \begin{cases} 0, & \text{si } (K_{min} - y_j) \leq 0 \\ (K_{min} - y_j)^{\frac{1}{y_j}} * (\Delta_T - \Delta_A)^{\frac{1}{\Delta_A}}, & \text{en otro caso} \end{cases} \quad (3.12)$$

**Sin considerar el tiempo:**

$$f_{ijq} = \begin{cases} 0, & \text{si } K_{min} - y_j \leq 0 \\ (K_{min} - y_j)^{\frac{1}{y_j}}, & \text{en otro caso} \end{cases} \quad (3.13)$$

En las restricciones 3.12 y 3.13 ya se hace uso del sub índice  $q$  mencionado en la sección 3.4.1, dado que las actividades generan mayor estrés si el puntaje acumulado ( $y_j$ ) es mucho menor al puntaje mínimo ( $K_{min}$ ) por lo que se deben colocar las actividades en la posición que genere menos estrés.

### 3.4.3 MODELACIÓN DE LA CALIDAD DEL MATERIAL

Otra característica considerada es la que se observa en la ecuación 3.14, que representa la calidad del material  $g_{ij}$  de la actividad  $i$  del subtema  $j$ , ya que a menor calidad en el material utilizado, mayor dificultad presentara el estudiante para completar una actividad.

#### Definición de Variables

$G_{min}$  .- Calidad mínima del material de una actividad.

$g_{ij}$  .- Calidad del material de la actividad  $i$  del subtema  $j$ .

$h_{ij}$  .- Que tanto afecta la calidad del material de la actividad  $i$  del subtema  $j$ .

$$h_{ij} = \begin{cases} 0, & \text{si } g_{ij} \geq G_{max} \\ \frac{G_{max}}{g_{ij}}, & \text{en otro caso} \end{cases} \quad (3.14)$$

### 3.4.4 COMPLEJIDAD DE LAS ACTIVIDADES

La ultima restricción representa la complejidad de una actividad ( $c_{ij}$ ) afectada por la calidad del material ( $h_{ij}$ ) y el estrés generado ( $f_{ijq}$ ), en una posición  $q$  en particular 3.15.

#### Definición de Variables

$z_{ijq}$  .- Complejidad de la actividad  $i$  del subtema  $j$  en la posición  $q$  afectada por el estrés y la calidad del material.

$c_{ij}$  .- Complejidad de la actividad  $i$  del subtema  $j$ .

$h_{ij}$  .- Que tanto afecta la complejidad de la actividad  $i$  del subtema  $j$  según la calidad del material.

$f_{ijq}$  .- Que tanto afecta la complejidad de la actividad  $i$  del subtema  $j$  según el estrés generado por dicha actividad en la posición  $q$ .

$$z_{ijq} = c_{ij} + f_{ijq} + h_{ij} \quad i = 1, 2, \dots, \eta; j = 1, 2, \dots, \alpha; q = 1, 2, \dots, \beta. \quad (3.15)$$

### 3.4.5 MODELACIÓN DE LA CAPACIDAD DEL ESTUDIANTE

La capacidad intelectual de un individuo se puede obtener mediante pruebas de medición de IQ como la prueba *WISE* (*Wechsler Intelligence Scale for Adults*) que nos permite saber aproximadamente la capacidad intelectual de un adulto y así asignarle un valor numérico, la cual definiremos como  $C_{max}$ .

Por lo que, la suma de las complejidades de las actividades ( $z_{ij}$ ) consideradas en un plan, afectadas por factores como el estrés y la calidad del material, tiene que ser menor o igual que la capacidad intelectual de un individuo ( $C_{max}$ ) 3.16.

$$\sum_{i=1}^{\eta} \sum_{j=1}^{\alpha} z_{ij} x_{ijq} \leq C_{max} \quad q = 1, 2, \dots, \beta. \quad (3.16)$$

### 3.4.6 MODELO MATEMÁTICO COMPLETO

A continuación se describe el modelo matemático completo, que nos proporciona que conjunto de actividades se deben incluir y en que orden deben de estar para que se obtenga el tiempo mínimo para aprobar un conjunto de materias con una calificación mínima, considerando factores como el estrés, la calidad del material, así como la capacidad del estudiante y una complejidad inicial para cada actividad.

#### Definición de Variables

$d_{ij}$  .- Duración de la actividad  $i$  del subtema  $j$ .

$u_{ij}$  .- Valor de la duración  $i$  del subtema  $j$ .

$x_{ijq}$  .- Variable binaria para denotar que se agrego la actividad  $i$  del subtema  $j$  en la posición  $q$ .

$c_{ij}$  .- Complejidad de la actividad  $i$  del subtema  $j$ .

$g_{ij}$  .- Calidad del material de la actividad  $i$  del subtema  $j$ .

$C_{max}$  .- Capacidad Intelectual máxima del estudiante.

$K_{max}$  .- Calificación máxima.

$K_{min}$  .- Calificación mínima.

$f_{ijq}$  .- Que tanto afecta el estrés a la complejidad de la actividad  $i$  del subtema  $j$  en la posición  $q$ .

$h_{ij}$  .- Que tanto afecta la calidad del material a la complejidad de la actividad  $i$  del subtema  $j$ .

$z_{ijq}$  .- Complejidad de la actividad  $i$  del subtema  $j$  afectada por el estrés y la calidad del material en la posición  $q$ .

#### Función Objetivo

$$\text{mín} \sum_{i=1}^{\eta} \sum_{j=1}^{\alpha} \sum_{q=1}^{\beta} d_{ij} x_{ijq}. \quad (3.17)$$

#### Sujeto a:

$$\sum_{i=1}^{\eta} \sum_{j=1}^{\alpha} u_{ij} x_{ijq} \geq K_{min} \quad q = 1, 2, \dots, \beta. \quad (3.18)$$

$$\sum_{i=1}^{\eta} \sum_{j=1}^{\alpha} u_{ij} x_{ijq} \leq K_{max} \quad q = 1, 2, \dots, \beta. \quad (3.19)$$

$$x_{ijq} \in 0, 1 \quad i = 1, 2, \dots, \eta; j = 1, 2, \dots, \alpha; q = 1, 2, \dots, \beta. \quad (3.20)$$

$$y_j = \sum_{i=1}^{\eta} \sum_{j=1}^{\alpha} u_{ij} x_{ijq} \quad q = 1, 2, \dots, \beta. \quad (3.21)$$

$$y_j \geq 0 \quad j = 1, 2, \dots, \alpha. \quad (3.22)$$

$$x_{i'j'q'} \leq \omega_{ii'} x_{ijq} \quad i, i' = 1, 2, \dots, \eta; j, j' = 1, 2, \dots, \alpha; q, q' = 1, 2, \dots, \beta. \quad (3.23)$$

$$x_{ij} = 1 \quad \forall_{m_{ij}} \in M. \quad (3.24)$$

$$h_{ij} = \begin{cases} 0, & si \quad g_{ij} \geq G_{max} \\ \frac{G_{min}}{g_{ij}}, & en \text{ otro caso} \end{cases} \quad (3.25)$$

$$f_{ijq} = \begin{cases} 0, & si \quad K_{min} - y_j \leq 0 \\ (K_{min} - y_j)^{\frac{1}{y_j}}, & en \text{ otro caso} \end{cases} \quad (3.26)$$

$$z_{ijq} = c_{ij} + f_{ijq} + h_{ij} \quad i = 1, 2, \dots, \eta; j = 1, 2, \dots, \alpha; q = 1, 2, \dots, \beta. \quad (3.27)$$

$$\sum_{i=1}^{\eta} \sum_{j=1}^{\alpha} z_{ij} x_{ijq} \leq C_{max} \quad q = 1, 2, \dots, \beta. \quad (3.28)$$

$$\sum_{i=1}^{\eta} \sum_{j=1}^{\alpha} x_{ijq} \leq 1 \quad q = 1, 2, \dots, \beta. \quad (3.29)$$

## CAPÍTULO 4

# EXPERIMENTACIÓN

---

Para la experimentación se hizo uso de las instancias descritas en la tabla 4.1 que fueron las utilizadas en el trabajo presentado por Maya P. et al. [9], estas instancias son las que representarían más desafío para el constructivo y el planificador ya que no solo se consideran problemas con una sola materia si no con más de una, para instancias pequeñas como problemas con 10 actividades y una materia, el planificador *OPTIC* alcanza el valor óptimo, es por ello que se optó por usar solo aquellas donde se pueda ver reflejado una mejora.

La experimentación se divide en dos secciones o fases, la primera corresponde con la generación de soluciones mediante el *CSA* para medir el makespan y la calidad de la misma, para después hacer una comparación con el valor óptimo y el valor obtenido con el planificador *OPTIC*, midiendo también su tiempo de cómputo y en el caso de *OPTIC* la cantidad de estados visitados durante la búsqueda.

Para la segunda parte de la experimentación se usaron las mejores soluciones obtenidas con el constructivo y se utilizaron como solución inicial para el planificador, esto debería generar mejores soluciones en el tiempo establecido de 60 minutos o encontrar soluciones de la misma calidad pero en menos tiempo.

La evaluación de las instancias se realizó en una laptop Lenovo Thinkpad con un procesador *Intel Core i7-4710MQ* y *32 gb RAM*, con un sistema operativo *Ubuntu* 15.10, el constructivo se desarrolló en lenguaje *R* en su versión 3.4.4.

## 4.1 DESCRIPCIÓN DE LAS INSTANCIAS

Las instancias usadas son un total de 95 problemas de los cuales se dividen en siete subcategorías dependiendo de la cantidad de materias que tenía cada problema, una descripción más completa en cuanto a la cantidad de materias, temas, subtemas y actividades es la que se puede observar en la tabla 4.1.

Tipo de instancias						
Instancia	Mat	Temas	Subt	Act	Cant. de Problemas	Total Act.
Grande-1	1	5	6	5	15	150
Grande-2	2	5	6	5	15	300
Grande-3	3	5	6	5	15	450
Grande-4	4	5	6	5	15	600
Grande-5	5	5	6	5	15	750
Grande-6	6	5	6	5	15	900
Grande-7	7	5	6	5	5	1050

Tabla 4.1: Cantidad de problemas por tipo de instancia según la cantidad de materias. La cantidad de temas es por materia, igual sucede con los subtemas, la cantidad es por cada tema y lo mismo las actividades, es la cantidad de actividades por subtema, al final se muestra el total de actividades por tipo de instancia.

El tipo de problema más grande sería el que cuenta con el mayor número materias y por ende de actividades, con un total de 1050 actividades, lo que generaría un espacio de búsqueda muy grande.

También se puede clasificar los problemas dependiendo de que tanta dependencia existe entre una actividad y otra, la cantidad de materias obligatorias que existen dentro de la instancia y el porcentaje de actividades con uno o dos requisitos, esto se observa en la tabla 4.2.

Instancias					
Tipo	%Dependencia	%Tipo A	%Tipo B	%Obligatorias	Cant. de Problemas
1	0	0	0	0	5
2	80	20	80	10	5
3	80	80	20	10	5

Tabla 4.2: Tipos de instancias según su dependencia, los valores en las celdas corresponden a porcentaje de actividades, El tipo *A* hace referencia a que solo requiere una actividad como requisito, tipo *B*, requiere dos actividades, la columna de obligatorias hace referencia al porcentaje de actividades que se tienen que incluir en la secuencia.

## 4.2 EXPERIMENTACIÓN 1

El objetivo de esta experimentación es comparar los tiempos de cómputo y la calidad de las soluciones, para ver la diferencia con respecto al valor obtenido con el *CSA* y con *OPTIC*.

Para el *CSA* se varió la cantidad de soluciones a generar, esto con el fin de verificar que existe una mejora de la calidad al incrementar el número de soluciones y ver que tanto afecta el tiempo de cómputo, se generaron 20000, 50000 y 100000 soluciones por instancia, de las cuales se extrae la mejor encontrada.

No se impuso un limite de tiempo para el constructivo ya que se espera sea mucho menor a *OPTIC*.

Para *OPTIC* se estableció un tiempo limite de 60 minutos para encontrar una solución.

### 4.2.1 RESULTADOS 1

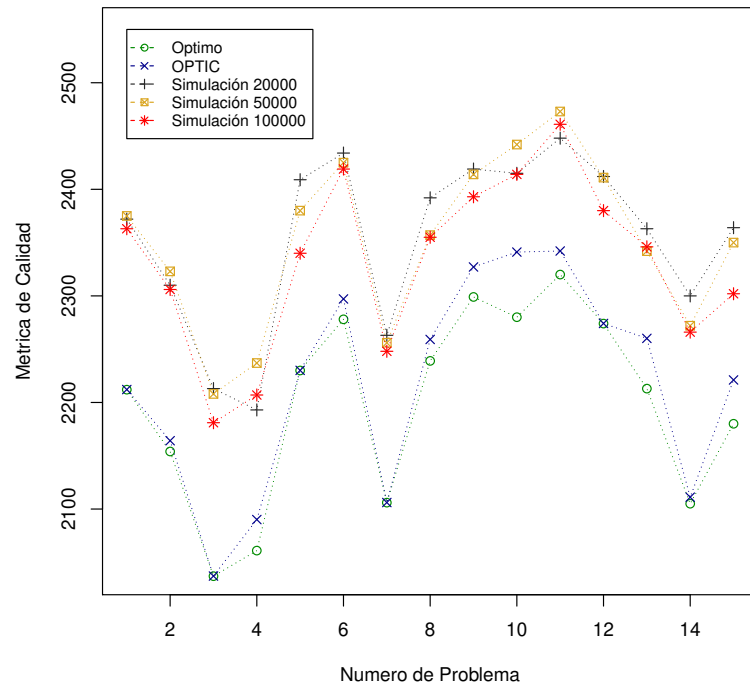
En la tabla 4.3 se puede apreciar la cantidad de problemas por instancia que fue capaz de resolver el planificador en el tiempo limite de 60 minutos. Y claramente se puede observar que conforme se incrementa la cantidad de materias, es más complicado para el planificador resolver el problema, esto debido al incremento del tamaño del espacio de búsqueda.

Instancias Resueltas				
Instancia	No Resueltos	Resueltos	%Sin Resolver	%Resuelto
1	0	15	0	100
2	1	14	7	93
3	2	13	14	86
4	9	6	60	40
5	13	2	87	13
6	15	0	100	0
7	5	0	100	0

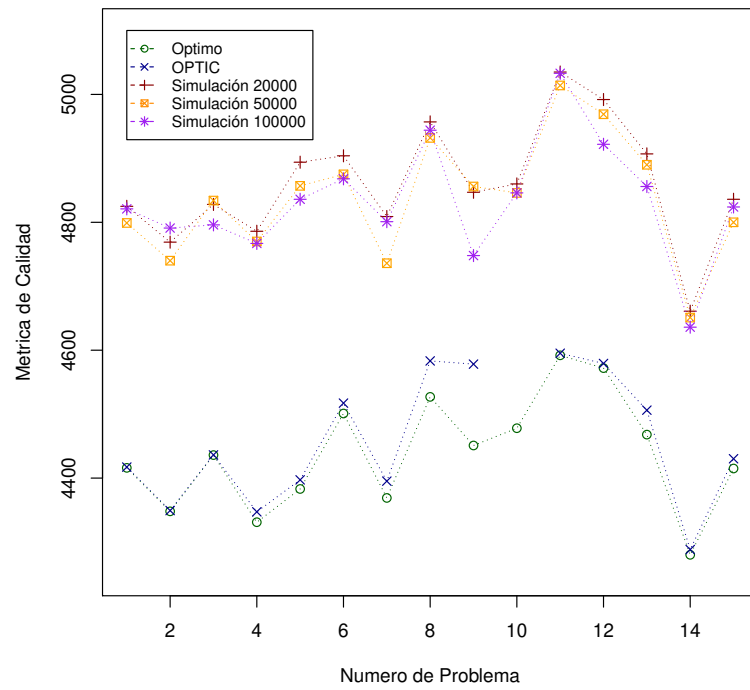
Tabla 4.3: Cantidad de instancias resueltas por *OPTIC*.

Como se puede observar en la figura 4.1 (inciso *a* y *b*) cuando se evalúa con dos materias o menos el planificador *OPTIC* obtiene resultados, muy cercanos al valor óptimo.

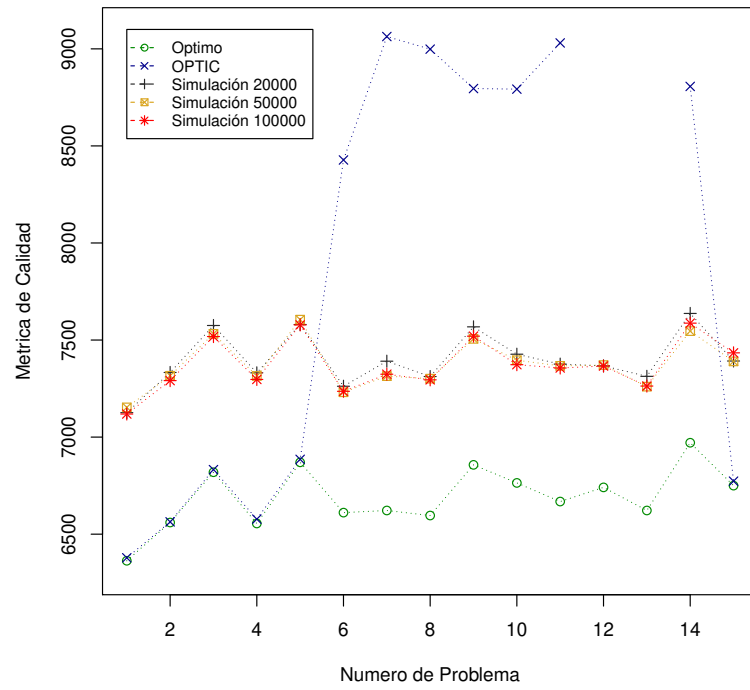




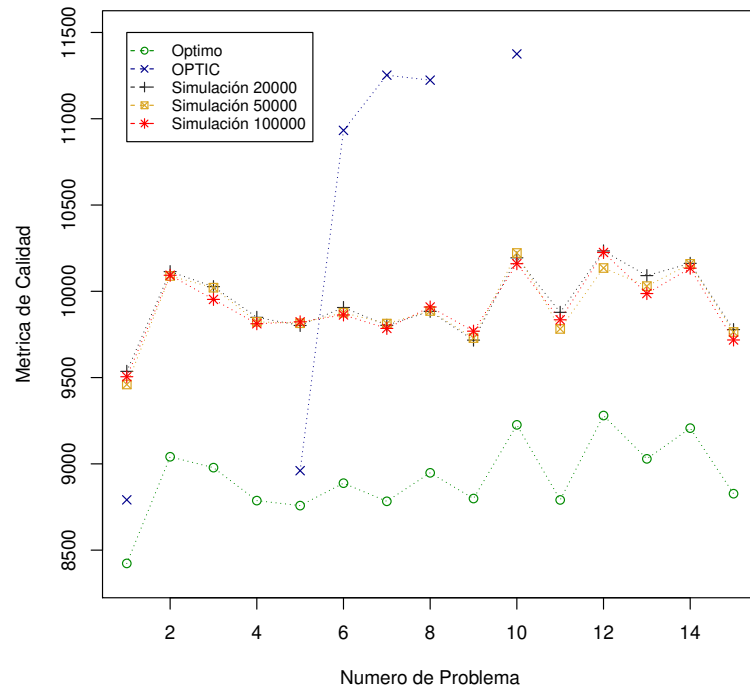
(a) Una materia



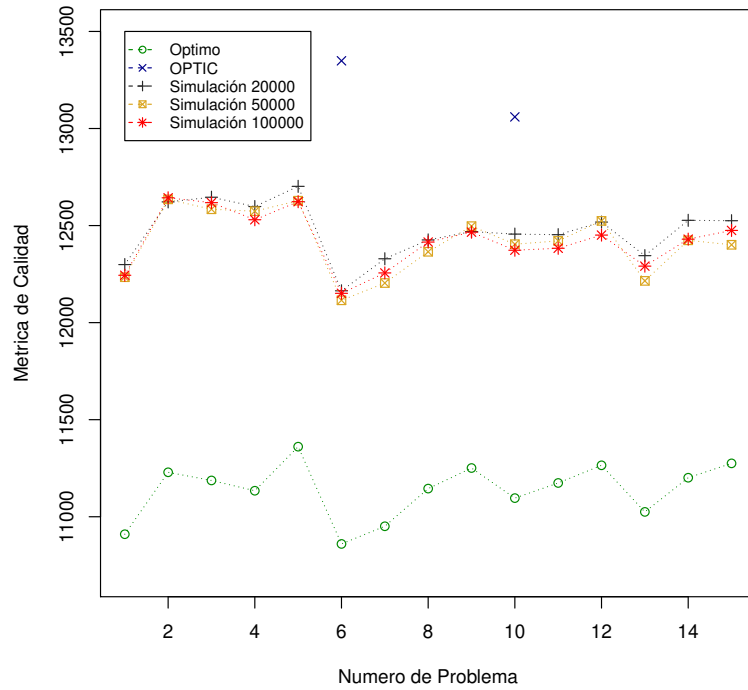
(b) Dos materias



(c) Tres materias



(d) Cuatro materias



(e) Cinco materias

Figura 4.1: Comparación de resultados obtenidos con el *CSA* y *OPTIC* contra el valor Óptimo, separados por la cantidad de materias.

Cuando se tienen pocas materias en un problema, *OPTIC* obtiene muy buenos resultados, pero en la figura 4.1 (inciso *c*, *d* y *e*), cuando se tiene tres o más materias, el planificador deja de ser muy eficiente esto debido a que su espacio de búsqueda se vuelve muy grande lo cual, al menos en el tiempo limite que se le impuso de 60 minutos, no logro encontrar una solución, mientras que el *CSA* se mantiene constante.

En la figura 4.2 se observan que los tiempos de computo del *CSA* están muy por debajo de los tiempos del planificador *OPTIC* como se esperaba ya que *OPTIC* utilizó en más del 50% de las instancias más de tres minutos para encontrar una solución, mientras que el constructivo tardo menos de un minuto, aun cuando se generan 100000 soluciones factibles, figura 4.3.

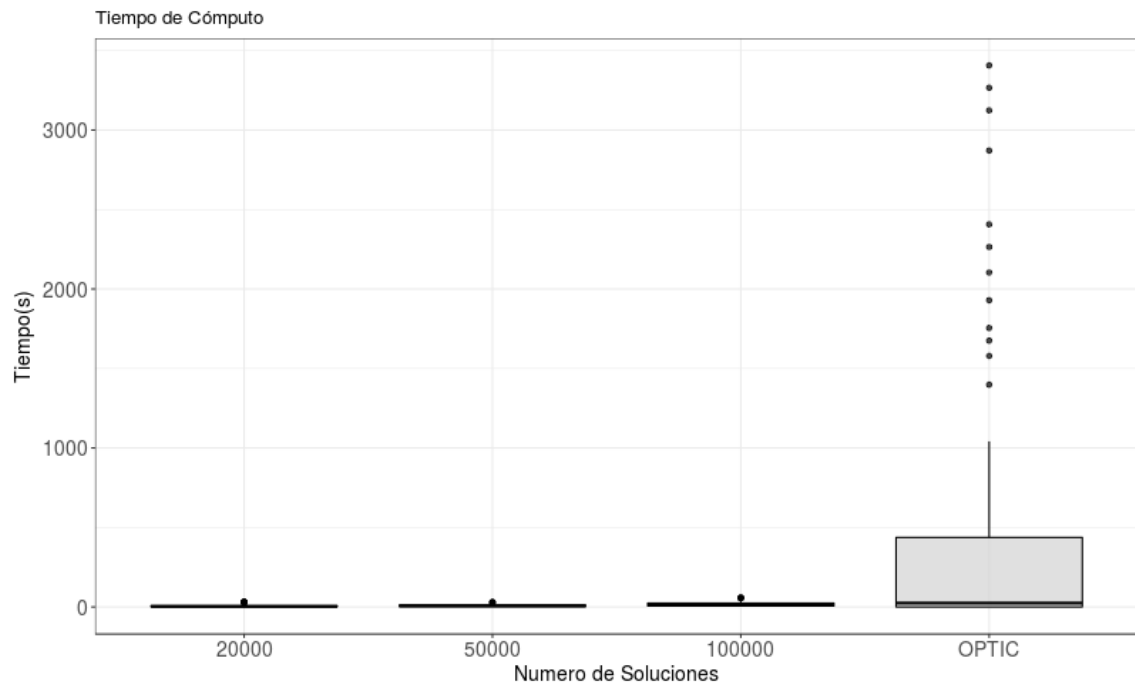


Figura 4.2: Comparativa de tiempos de computo entre OPTIC y el constructivo.

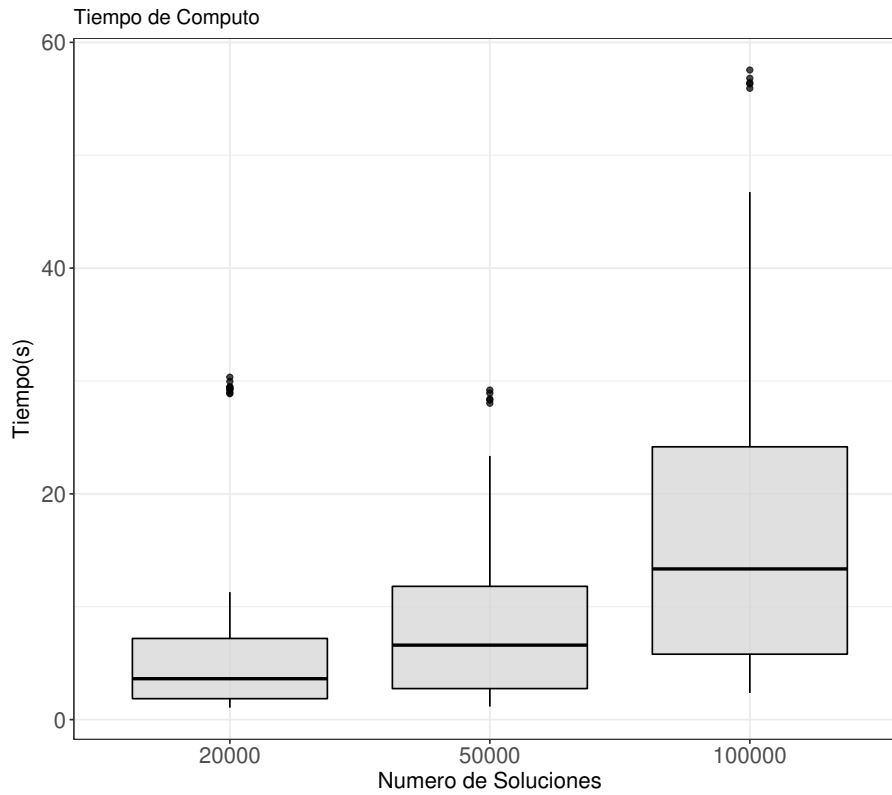


Figura 4.3: Tiempo de computo del constructivo al aumentar la cantidad de soluciones a generar.

La comparativa del *gap* al aumentar la cantidad de soluciones generadas a 20000, 50000 y 100000 es la que se puede observar en la figura 4.4 y se puede decir que es estable, conforme aumentamos la dificultad del problema el *gap* aumenta ligeramente mientras que *OPTIC* no alcanza a resolver todos los problemas con más de una materia, es por ello que solo aparece un punto cuando solo se tiene una materia que fue el único tipo de instancia en la cual resolvió todo.

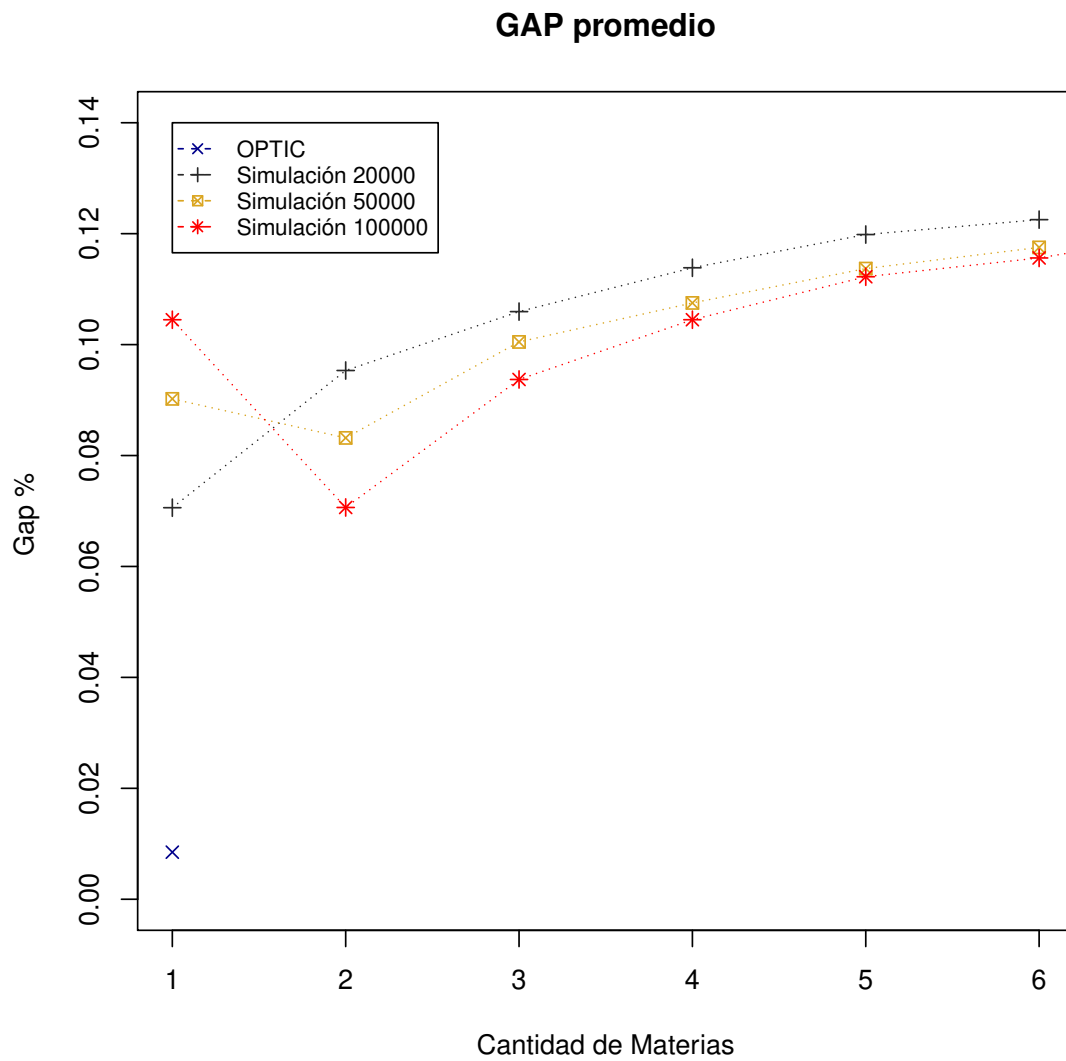


Figura 4.4: Gap promedio del constructivo y el planificador OPTIC.

Y como se ve en la figura 4.5 el *gap* se encuentra alrededor del 10 % del valor óptimo.

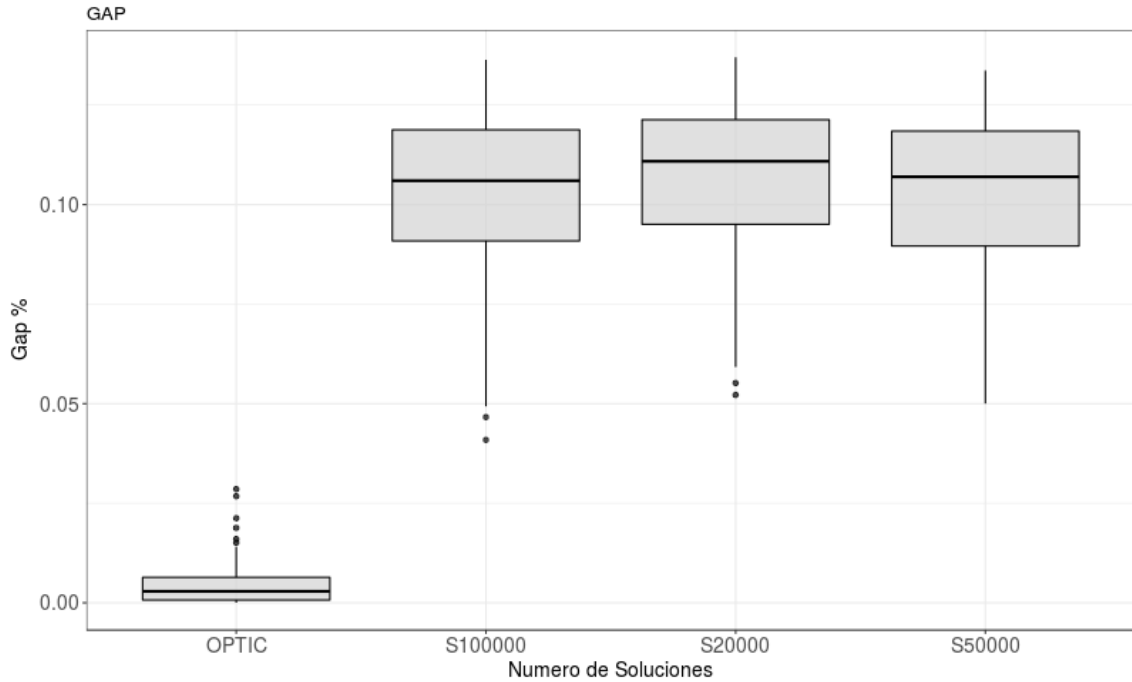


Figura 4.5: Gap promedio en caja de bigotes.

### 4.3 EXPERIMENTACIÓN 2

Para la segunda parte de la experimentación se busca que, una vez obtenido resultados con el constructivo, ver que tanta mejora existe si las soluciones encontradas sirven como solución inicial para el planificador y dado que *OPTIC* dejó sin resolver problemas con más de dos materias, validar si al darle una solución inicial puede mejorarla o encontrar esa solución en el tiempo límite que es de 60 minutos y así poder definir el área en la que *OPTIC* es capaz de encontrar una solución.

Las soluciones utilizadas fueron las obtenidas al generar 100000 soluciones con el constructivo ya que son las de mejor calidad.

### 4.3.1 RESULTADOS 2

Como se observa en la figura 4.6, alrededor de un 25 % de las instancias obtuvieron una mejora en la calidad de la solución mediante el uso de soluciones iniciales generadas por el *CSA*. Incluso, se pudo obtener solución a problemas en los que el planificador *OPTIC* no pudo encontrar en 60 minutos por si solo.

Los puntos que no aparece en la gráfica de la figura 4.6 representan aquellas instancias en las que ni proporcionándole una solución inicial, se logro encontrar una solución mediante el planificador en el tiempo fijado de 60 minutos.

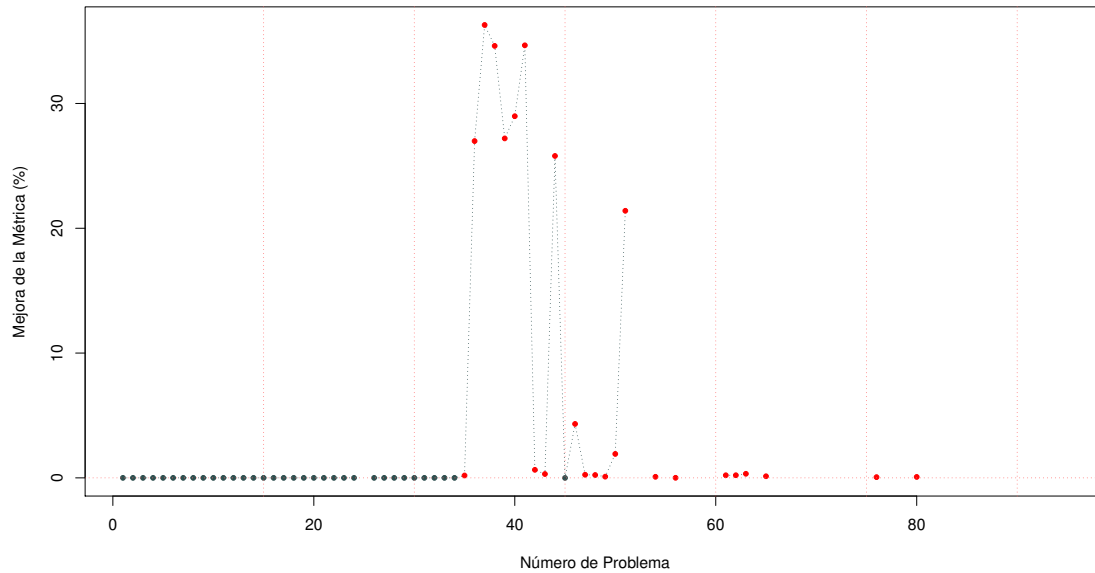


Figura 4.6: Instancias en las que se mejoro la solución encontrada con *OPTIC* utilizando soluciones iniciales proporcionadas por el *CSA* (puntos en color rojo).

La tabla 4.4 muestra exactamente cuales fueron las instancias que presentaron un mejor beneficio al proporcionar soluciones iniciales. En la mayoría de los casos ocurrió a partir de dos materias, siendo las de tres y cuatro las que tuvieron una mejora más significativa.



Instancias que mejoraron	
Instancia	Cuántas Mejoraron
1	0
2	0
3	10
4	8
5	4
6	2
7	0

Tabla 4.4: Cantidad de instancias que mejoraron sus soluciones *OPTIC*.

En algunos casos no se logró mejorar la solución encontrada con el planificador utilizando soluciones iniciales. Esto debido a que, se logró encontrar el valor óptimo sin usar la solución proporcionada por el *CSA* o el *gap* estaba por debajo del 3% con respecto al valor óptimo.

También, en la figura 4.7 se observa que el *gap* disminuyó considerablemente al utilizar una solución inicial, alrededor de un 20% en todos los tamaños de instancias, exceptuando cuando se tiene una o dos materias, ya que como se mencionó, *OPTIC* está muy cerca o en el valor óptimo.

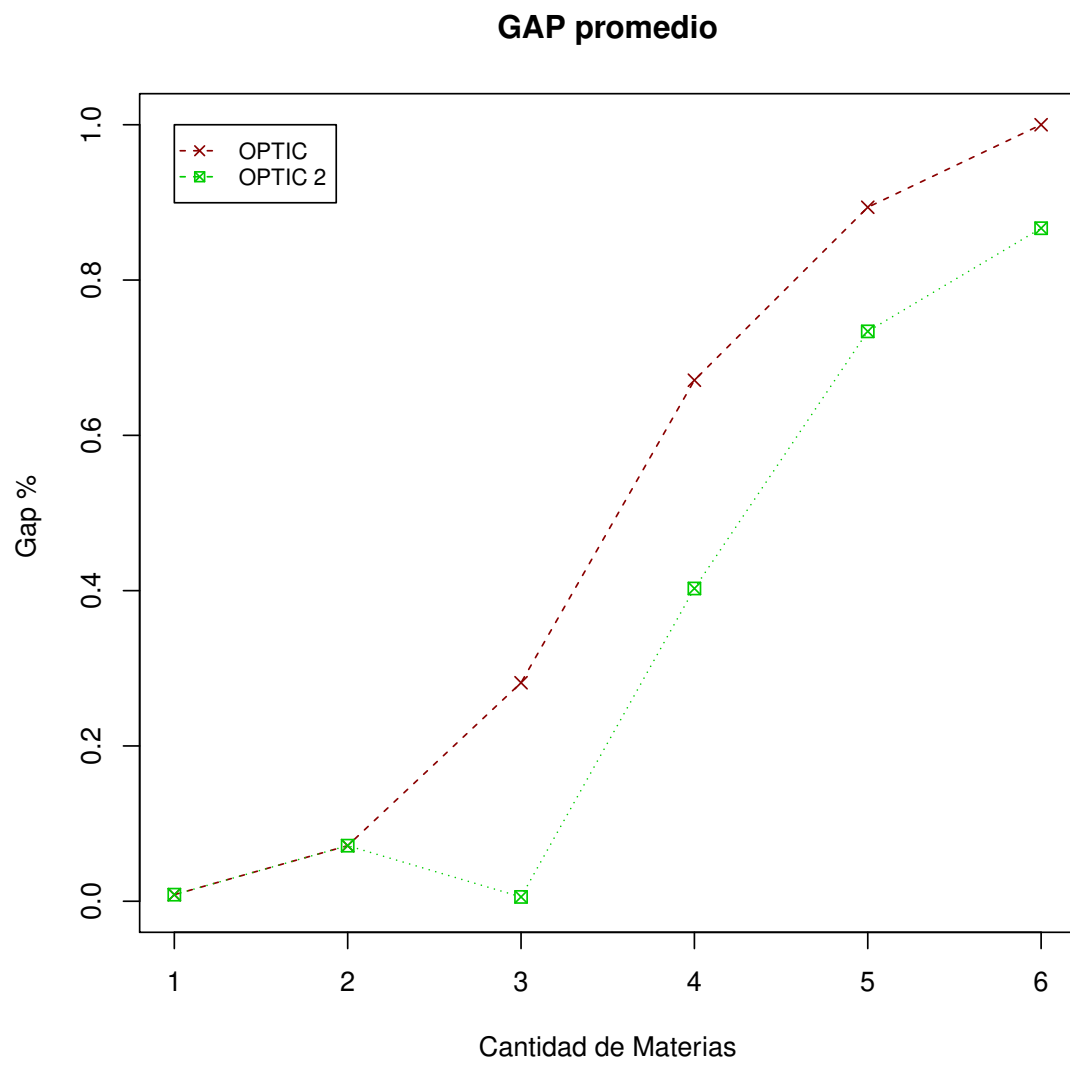


Figura 4.7: Gap promedio.

## CAPÍTULO 5

# CONCLUSIONES

---

En este trabajo se presento un algoritmo constructivo el cual genera soluciones de buena calidad, las cuales funcionan como soluciones iniciales para reducir el espacio de búsqueda de un planificador automatizado y lograr encontrar soluciones de igual o mejor calidad que como lo hace el planificador por si mismo.

Además se extendió el modelo matemático del PPE para lograr generar rutas de aprendizaje que pueden ser afectadas por situaciones como el estrés o la capacidad intelectual del estudiante, volviendo el problema más apegado al mundo real.

## 5.1 CONTRIBUCIONES

Una de las principales contribuciones fue el desarrolló de un algoritmo constructivo (CSA) que nos permite generar soluciones de buena calidad y rápidamente. Las soluciones no son óptimas, presentan un *gap* del 10 % con respecto al valor óptimo, pero tiene un comportamiento estable sin importar el tamaño de las instancias.

Para las instancias de mayor tamaño (más de dos materias), el *gap* presentado por las soluciones generadas por el *CSA* fue mejor que el *gap* de las soluciones encontradas por el planificador por si mismo.

Lo que sugiere que es una buena opción usar las soluciones generadas por el *CSA* como soluciones iniciales para el planificador, ya que se mejoró en muchos casos las soluciones encontradas por *OPTIC*. En algunos casos se encontró solución

a problemas en los que el planificador no lo hizo en el tiempo límite de 60 minutos.

Se presento también, una extensión al modelo matemático presentado por Maya P. et al. [9] para el *PPE*, al cual se le añadieron restricciones que le permiten considerar características subjetivas como el estrés, la capacidad intelectual del estudiante y la calidad del material. Estas nuevas restricciones afectan en la selección de actividades y el orden en el que estarían dispuesta en el plan generado.

Al incluir este tipo de restricciones, nos permite apegar el problema más al mundo real y por lo tanto, generar rutas de aprendizaje personalizadas.

## CAPÍTULO 6

# TRABAJO FUTURO

---

Como se menciono anteriormente una de las áreas de investigación de la aplicación de sistemas inteligentes en la educación es poder incluir aspectos subjetivos dentro del modelo matemático y que sean considerados para la selección de las actividades por ello como trabajo a futuro se pretende lo siguiente:

- Generar soluciones mediante otro algoritmo para comparar la calidad de las soluciones y el tiempo de computo con el *CSA*, se espera que al usar cualquier otro método constructivo como un *GRASP* o alguna otra meta-heurística, el tiempo de computo sea mayor al del *CSA*.
- Extender el modelo matemático para considerar otros aspectos personales como lo son la edad, la disposición del estudiante para trabajar, etc., esto para volver más real y personal, la selección de actividades.
- Una vez generado el modelo lo siguiente sería modelar las nuevas restricciones en *PDDL* para poder ser ejecutadas por los planificadores.
- Generar instancias considerando esas nuevas restricciones, actualmente hay aspectos como el estrés o la capacidad intelectual de los cuales ya existen estudios psicológicos como el WAIS [10,13] para medir la capacidad intelectual en adultos en los cuales les asignan un valor numérico lo cual sería de gran utilidad para poder dar un peso a las actividades.

- 
- Realizar una comparativa entre el valor óptimo y la solución generada por un planificador considerando esas nuevas restricciones.

## APÉNDICE A

# EJEMPLOS

---

### A.1 PDDL

Problema y dominio del problema usado como ejemplo en este trabajo.

## A.1.1 PROBLEMA PDDL

---

```

1: (define (problem degree-example)
2:   (:domain degree)
3:   (:objects
4:     student1 - student
5:   )
6:   (:init
7:     (free student1)
8:     (= (total-credits-subject-gain student1) 0)
9:     (= (available-credits student1) 48)
10:    (available-subject Material1 student1)
11:    (= (credits-subject Material1) 9)
12:    (= (mingrade Material1) 70)
13:    (not-approved Material1 student1)
14:    (not-done-LA LA1 Material1 student1)
15:    (not-done-LA LA2 Material1 student1)
16:    (not-done-LA LA3 Material1 student1)
17:    (not-done-LA LA4 Material1 student1)
18:    (not-done-LA LA5 Material1 student1)
19:    (= (score Subtema1 student1) 0)
20:    (not-done-LA LA6 Material1 student1)
21:    (not-done-LA LA7 Material1 student1)
22:    (not-done-LA LA8 Material1 student1)
23:    (not-done-LA LA9 Material1 student1)
24:    (not-done-LA LA10 Material1 student1)
25:    (= (score Subtema2 student1) 0)
26:    (= (quantity-resource rec0) 100000)
27:    (= (quantity-resource rec1) 20)
28:    (= (quantity-resource rec2) 30)
29:    (= (quantity-resource rec3) 40)
30:    (= (valueLA LA1) 15)
31:    (= (valueLA LA2) 22)
32:    (= (valueLA LA3) 21)
33:    (= (valueLA LA4) 15)
34:    (= (valueLA LA5) 27)
35:    (= (valueLA LA6) 19)
36:    (= (valueLA LA7) 22)
37:    (= (valueLA LA8) 23)
38:    (= (valueLA LA9) 23)
39:    (= (valueLA LA10) 13)
40:    (= (DurationLA LA1) 5)
41:    (= (DurationLA LA2) 36)
42:    (= (DurationLA LA3) 26)
43:    (= (DurationLA LA4) 7)
44:    (= (DurationLA LA5) 30)
45:    (= (DurationLA LA6) 6)
46:    (= (DurationLA LA7) 31)
47:    (= (DurationLA LA8) 34)
48:    (= (DurationLA LA9) 23)
49:    (= (DurationLA LA10) 13)

```

---



---

```

50: (isPartOfSubtheme LA1 Subtema1)
51: (isPartOfSubtheme LA2 Subtema1)
52: (isPartOfSubtheme LA3 Subtema1)
53: (isPartOfSubtheme LA4 Subtema1)
54: (isPartOfSubtheme LA5 Subtema1)
55: (isPartOfSubtheme LA6 Subtema2)
56: (isPartOfSubtheme LA7 Subtema2)
57: (isPartOfSubtheme LA8 Subtema2)
58: (isPartOfSubtheme LA9 Subtema2)
59: (isPartOfSubtheme LA10 Subtema2)
60: (isPartOfTheme Subtema1 Tema1)
61: (isPartOfTheme Subtema2 Tema1)
62: (isPartOfSubject Tema1 Material1)
63: (KindResourceLO LA1 rec3)
64: (KindResourceLO LA2 rec1)
65: (KindResourceLO LA3 rec3)
66: (KindResourceLO LA4 rec2)
67: (KindResourceLO LA5 rec3)
68: (KindResourceLO LA6 rec3)
69: (KindResourceLO LA7 rec1)
70: (KindResourceLO LA8 rec1)
71: (KindResourceLO LA9 rec3)
72: (KindResourceLO LA10 rec1)
73: (not-has-reqs LA1)
74: (not-has-reqs LA2)
75: (not-has-reqs LA3)
76: (not-has-reqs LA4)
77: (not-has-reqs LA5)
78: (not-has-reqs LA6)
79: (not-has-reqs LA8)
80: (not-has-reqs LA9)
81: (has-reqs LA7 LA5)
82: (has-multiple-reqs LA10 LA4)
83: (has-multiple-reqs LA10 LA2)
84: (= (maxgrade-subtheme Subtema1) 100)
85: (= (maxgrade-subtheme Subtema2) 100)
86: )
87: ;;objetivo: termine todas las materias
88: (:goal (and
89: (pass-degree Material1 student1)
90: )
91: )
92: ;; metrica a optimizar
93: (:metric minimize (total-time))
94: ;;fin del problema
95: )

```

---

## A.1.2 DOMINIO PDDL

---

```

1: (define (domain degree)
2: (:requirements :durative-actions :typing :fluents :equality)
3: (:types student resource - object
4:  subject Theme subtheme LA - LO)
5: (:constants
6:  Material - subject
7:  Tema1 - Theme
8:  Subtema1
9:  Subtema2 - subtheme
10: LA1
11: LA2
12: LA3
13: LA4
14: LA5
15: LA6
16: LA7
17: LA8
18: LA9
19: LA10 - LA
20: rec0
21: rec1
22: rec2
23: rec3 - resource
24: )
25: (:predicates
26:  (available-subject ?subj - subject ?s - student)
27:  (free ?s - student)
28:  (pass-degree ?subj - subject ?s - student)
29:  (enrollment ?s - student ?subj - subject)
30:  (done-Theme ?t - Theme ?subj -subject ?s - student)
31:  (done-subject-LA ?subj - subject ?s - student)
32:  (not-done-LA ?oa - LA ?subj - subject ?s - student)
33:  (not-approved ?subj - subject ?s - student)
34:  (isPartOfSubtheme ?oa - LA ?subt - subtheme)
35:  (isPartOfTheme ?subt - subtheme ?t - Theme)
36:  (isPartOfSubject ?t - Theme ?subj - subject)
37:  (KindResourceLO ?oa - LA ?eq - resource)
38:  (not-has-reqs ?oa - LA)
39:  (has-reqs ?oa - LA ?req - LO)
40:  (has-multiple-reqs ?oa - LA ?req - LO)
41:  (done ?oa - LA)
42: )
43: (:functions
44:  (credits-subject ?subj - subject)
45:  (total-credits-subject-gain ?s - student)
46:  (available-credits ?s - student)
47:  (score ?req - LO ?s - student)
48:  (quantity-resource ?eq - resource)
49:  (valueLA ?oa - LA)
50:  (mingrade ?subj - subject)
51:  (DurationLA ?oa - LA)
52:  (maxgrade-subtheme ?subt - subtheme)
53:  (amount-in-subtheme ?oa - LA)
54: )

```

---

---

```

55: (:durative-action enroll - subjectMaterial
56: :parameters (?s - student)
57: :duration (= ?duration 0)
58: :condition (and
59:   (at start (available-subject Material ?s))
60:   (at start (not-approved Material ?s))
61:   (at start (<(credits-subject Material1)(available-credits ?s)))
62: )
63: :effect (and
64:   (at end (enrollment ?s Material1))
65:   (at end (decrease (available-credits ?s)(credits-subject Material1)))
66:   (at end (not (available-subject Material1 ?s)))
67: )
68: )
69: (:durative-action CHOOSE-LA-nothasreqs
70: :parameters (?s - student ?oa - LA ?subt - subtheme ?t - Theme ?subj - subject ?eq -
71:   resource)
72: :duration (= ?duration (DurationLA ?oa))
73: :condition (and
74:   (at start (free ?s))
75:   (at start (enrollment ?s ?subj))
76:   (at start (not-done-LA ?oa ?subj ?s))
77:   (at start (isPartOfSubtheme ?oa ?subt))
78:   (at start (isPartOfTheme ?subt ?t))
79:   (at start (isPartOfSubject ?t ?subj))
80:   (at start (KindResourceLO ?oa ?eq))
81:   (at start (>(quantity-resource ?eq 0))
82:   (at start (not-has-reqs ?oa))
83:   (at start (>(maxgrade-subtheme ?subt)(valueLA ?oa)))
84: )
85: :effect (and
86:   (at start (not(free ?s)))
87:   (at start (decrease (quantity-resource ?eq 1))
88:   (at end (increase (quantity-resource ?eq 1))
89:   (at end (not (not-done-LA ?oa ?subj ?s)))
90:   (at end (increase (score ?subt ?s) (valueLA ?oa)))
91:   (at end (free ?s))
92:   (at end (decrease (maxgrade-subtheme ?subt)(valueLA ?oa)))
93:   (at end (done ?oa))
94: )
95: )
96: ;;//Clase 1. Un LA requiere un subtema (sea >0 ó >70) y a) anterior ya sea continuo o no
97: (:durative-action CHOOSE-LA-hasreqsSubtheme
98: :parameters (?s - student ?oa - LA ?subt - subtheme ?t - Theme ?subj - subject ?eq -
99:   resource ?req - LO)
100: :duration (= ?duration (DurationLA ?oa))
101: :condition (and
102:   (at start (free ?s))
103:   (at start (enrollment ?s ?subj))
104:   (at start (not-done-LA ?oa ?subj ?s))
105:   (at start (isPartOfSubtheme ?oa ?subt))
106:   (at start (isPartOfTheme ?subt ?t))
107:   (at start (isPartOfSubject ?t ?subj))
108:   (at start (KindResourceLO ?oa ?eq))
109:   (at start (>(quantity-resource ?eq 0))
110:   (at start (has-reqs ?oa ?req))
111:   (at start (>(score ?req ?s) (amount-in-subtheme ?oa)))
112:   (at start (>(maxgrade-subtheme ?subt)(valueLA ?oa)))
113: )

```

---

---

```

114: :effect (and
115:   (at start (not(free ?s)))
116:   (at start (decrease (quantity-resource ?eq) 1))
117:   (at end (increase (quantity-resource ?eq) 1))
118:   (at end (not (not-done-LA ?oa ?subj ?s)))
119:   (at end (increase (score ?subt ?s) (valueLA ?oa)))
120:   (at end (free ?s))
121:   (at end (decrease (maxgrade-subtheme ?subt)(valueLA ?oa)))
122:   (at end (done ?oa))
123: )
124: )
125: ;;//Clase 2. Un LA requiere otro LA
126: (:durative-action CHOOSE-LA-hasreqsLA
127: :parameters (?s - student ?oa - LA ?subt - subtheme ?t - Theme ?subj - subject ?eq -
128:   resource ?req - LA)
129: :duration (= ?duration (DurationLA ?oa))
130: :condition (and
131:   (at start (free ?s))
132:   (at start (enrollment ?s ?subj))
133:   (at start (not-done-LA ?oa ?subj ?s))
134:   (at start (isPartOfSubtheme ?oa ?subt))
135:   (at start (isPartOfTheme ?subt ?t))
136:   (at start (isPartOfSubject ?t ?subj))
137:   (at start (KindResourceLO ?oa ?eq))
138:   (at start (>(quantity-resource ?eq) 0))
139:   (at start (has-reqs ?oa ?req))
140:   (at start (done ?req))
141:   (at start (>(maxgrade-subtheme ?subt)(valueLA ?oa)))
142: )
143: :effect (and
144:   (at start (not(free ?s)))
145:   (at start (decrease (quantity-resource ?eq) 1))
146:   (at end (increase (quantity-resource ?eq) 1))
147:   (at end (not (not-done-LA ?oa ?subj ?s)))
148:   (at end (increase (score ?subt ?s) (valueLA ?oa)))
149:   (at end (free ?s))
150:   (at end (decrease (maxgrade-subtheme ?subt)(valueLA ?oa)))
151:   (at end (done ?oa))
152: )
153: )
154: ;;//clase 3 un LA req varios LA
155: (:durative-action CHOOSE-LA-hasreqs-multipleLA2
156: :parameters (?s - student ?oa - LA ?subt - subtheme ?t - Theme ?subj - subject ?eq -
157:   resource ?req1 - LA ?req2 - LA)
158: :duration (= ?duration (DurationLA ?oa))
159: :condition (and
160:   (at start (not(= ?req1 ?req2)))
161:   (at start (free ?s))
162:   (at start (enrollment ?s ?subj))
163:   (at start (not-done-LA ?oa ?subj ?s))
164:   (at start (isPartOfSubtheme ?oa ?subt))
165:   (at start (isPartOfTheme ?subt ?t))
166:   (at start (isPartOfSubject ?t ?subj))

```

---

---



---

```

167: (at start (KindResourceLO ?oa ?eq))
168: (at start (>(quantity-resource ?eq) 0))
169: (at start (done ?req1))
170: (at start (done ?req2))
171: (at start (has-multiple-reqs ?oa ?req1))
172: (at start (has-multiple-reqs ?oa ?req2))
173: (at start (>(maxgrade-subtheme ?subt)(valueLA ?oa)))
174: )
175: :effect (and
176: (at start (not(free ?s)))
177: (at start (decrease (quantity-resource ?eq) 1))
178: (at end (increase (quantity-resource ?eq) 1))
179: (at end (not (not-done-LA ?oa ?subj ?s)))
180: (at end (increase (score ?subt ?s) (valueLA ?oa)))
181: (at end (free ?s))
182: (at end (decrease (maxgrade-subtheme ?subt)(valueLA ?oa)))
183: (at end (done ?oa))
184: )
185: )
186: (:durative-action PASS-Theme-Tema1Material
187: :parameters (?s - student)
188: :duration (= ?duration 0)
189: :condition (and
190: (at start (enrollment ?s Material1))
191: (at start (>= (score Subtema1 ?s)(mingrade Material1)))
192: (at start (>= (score Subtema2 ?s)(mingrade Material1)))
193: )
194: :effect (and
195: (at end (done-Theme Tema1 Material1 ?s))
196: )
197: )
198: (:durative-action PASS-Material1
199: :parameters (?s - student)
200: :duration (= ?duration 0)
201: :condition (and
202: (at start (enrollment ?s Material1))
203: (at start (done-Theme Tema1 Material1 ?s))
204: )
205: :effect
206: (at end (done-subject-LA Material1 ?s))
207: )
208: (:durative-action take-subject-pass
209: :parameters (?s - student ?subj - subject)
210: :duration (= ?duration 0)
211: :condition (and
212: (at start (enrollment ?s ?subj))
213: (at start (done-subject-LA ?subj ?s))
214: )
215: :effect (and
216: (at end (not (not-approved ?subj ?s)))
217: (at end (increase (total-credits-subject-gain ?s) (credits-subject ?subj)))
218: (at end (not (available-subject ?subj ?s)))
219: (at end (pass-degree ?subj ?s))
220: )
221: )
222: )
223: ;;FIN DEL DOMINIO

```

---

# BIBLIOGRAFÍA

---

- [1] AMANDA COLES, M. F., ANDREW COLES y D. LONG, «COLIN Planning with Continuous Linear Numeric Change», *Department of Informatics, Kings College London, Strand, London WC2R 2LS, UK*, **1**(2), págs. 43–48, 2012.
- [2] ANDREW COLES, A. C. y J. BENTON, «Temporal Planning with Preferences and Time Dependent Continuous Costs», *Twenty-Second International Conference on Automated Planning and scheduling*, 2008.
- [3] ANDREW COLES, D. L., MARIA FOX y A. SMITH, «Planning with Problems Requiring Temporal Coordination», *Department of Computer and Information Sciences, University of Strathclyde, Glasgow, G1 1XH, UK*, **1**(2), págs. 43–48, 2008.
- [4] ANDREW COLES, D. L., MARIA FOX y A. SMITH, «Forward–Chaining Partial–Order Planning», *Twentieth International Conference on Automated Planning and Scheduling (ICAPS 2010)*, 2010.
- [5] BADER, D. A., W. E. HART y C. A. PHILLIPS, *Parallel Algorithm Design for Branch and Bound*, 2005.
- [6] CHEN, C. M., «Intelligent web based learning system with personalized learning path guidance», *Computers and Education*, **51**(2), págs. 787–814, 2008.
- [7] CHUANG, G.-J. H. F.-R. K. P.-Y. Y. K.-H., «A Heuristic Algorithm for planning personalized learning paths for context-aware ubiquitous learning», *Elsevier*, **54**(2), págs. 404–415, 2010.

- 
- [8] CRINA GROSAN, A. A., *Intelligent Systems*, Springer Verlag Berlin Heidelberg, 2011.
- [9] CRISTINA MAYA PADRÓN, R. S. N., «Generación de trayectorias de aprendizaje a través de modelos educativos de planificación», *Ciencia UANL*, págs. 56–61, 2001.
- [10] DENNIS, D. M., «Predicting Full Scale WAIS IQS With the Shipley Hartford», *The Clinical Neuropsychologist*, 1973.
- [11] EBRU OZPOLAT, G. B. A., «Automatic detection of learning styles for an elearning system», *Computers and Education*, **53**(2), págs. 355–367, 2009.
- [12] HOFFMANN, J., «A Heuristic for Domain Independent Planning and its Use in an Enforced Hill climbing Algorithm», *Operations Research Letters*, **1**(2), págs. 43–48, 1982.
- [13] J. R. CRAWFORD, K. M. A., «Estimating Premorbid WAIS RIQ with Demographic Variables: Regression Equations Derived from a UK Sample», *The Clinical Neuropsychologist*, **11**(2), págs. 192–197, 1997.
- [14] JORG HOFFMANN, B. N., «The FF Planning System: Fast Plan Generation Through Heuristic Search», *Journal of Artificial Intelligence Research*, **14**, págs. 253–302, 2001.
- [15] KNOBLOCK, C. A., «The Planning Domain Definition Language», *ResearchGate*, 1998.
- [16] MALIK GHALLAB, P. T., DANA NAU, *Automated Planning and Acting*, Cambridge University Press, 2016.
- [17] PETER BRUSILOVSKY, J. E. y E. SCHWARTZ, «Web based education for all a tool for develop adaptive courseware», *Computer Network and ISDN Systems*, **30**(2), págs. 291–300, 1998.

- 
- [18] RUBÉN PEREDO, A. M., ALEJANDRO CANALES y I. PEREDO, «Intelligent Webbased education system for adaptive learning», *Elsevier*, **38**(2), págs. 14 690–14 702, 2011.
- [19] RUSSELL, S. y P. NORVING, *Artificial Intelligence A Modern Approach*, tercera edición, Prentice Hall, Pearson, Upper Saddle River, New Jersey 07458, 2010.



## **FICHA AUTOBIOGRÁFICA**

Juan Pablo Rosas Baldazo

Candidato para el grado de Maestría en Ciencias en Ingeniería de Sistemas

Universidad Autónoma de Nuevo León  
Facultad de Ingeniería Mecánica y Eléctrica

Tesis: Evaluación del planificador automatizado OPTIC al generar rutas de aprendizaje mediante el uso de cotas superiores y extensión del modelo matemático.